



SUSE LINUX

管理指南

Edition 4 2005

Copyright ©

本出版品是Novell Inc. 的智慧財產。

內容可以部份或全部重製，條件是每一個複本上都要可以清楚看到版權標籤。

本手冊中所有資訊在編輯時，都已全力注意各項細節。但這不保證百分之百的正確性。因此，SUSE LINUX GmbH、作者或譯者都不需對任何錯誤或造成的結果負責。

本手冊中引用的很多軟體和硬體說明，皆為註冊商標。所有商業名稱皆受版權限制，而且可能是註冊商標。SUSE LINUX GmbH 十分注重製造商的拼法。本手冊出現的產品名稱和商標(帶有或不帶有特定標記)皆受商標和交易保護法管轄，因此受到版權的限制。

您若有任何建議和意見，請來信至<documentation@suse.de>。

Authors: Stefan Behlert, Frank Bodammer, Stefan Dirsch, Olaf Donjak, Roman Drahtmuller, Torsten Duwe, Thorsten Dubiel, Thomas Fehr, Stefan Fent, Werner Fink, Kurt Garloff, Carsten Gros, Joachim Gleisner, Andreas Grunbacher, Franz Hassels, Andreas Jaeger, Klaus Kampf, Andi Kleen, Hubert Mantel, Lars Marowsky-Bree, Chris Mason, Johannes Meixner, Lars Muller, Matthias Nagorni, Anas Nashif, Siegfried Olschner, Peter Poml, Thomas Renninger, Heiko Rommel, Marcus Schafer, Nicolaus Schuler, Klaus Singvogel, Hendrik Vogelsang, Klaus G. Wagner, Rebecca Walter, Christian Zoz

Translators: Daniel Pisano, Tino Tanner

Editors: Jorg Arndt, Antje Faber, Berthold Gunreben, Roland Haidl, Jana Jaeger, Edith Parzefall, Ines Pozo, Thomas Rolz, Thomas Schraitle, Rebecca Walter

Layout: Manuela Piotrowski, Thomas Schraitle

Setting: DocBook-XML, L^AT_EX

This book has been printed on 100% chlorine-free bleached paper.

歡迎使用

恭喜您使用新的Linux作業系統，同時也感謝您採用SUSE LINUX 9.3。購買此版本，您就可以依照<http://www.novell.com/products/linuxprofessional/support/conditions.html> 說明的方式，透過電話及電子郵件取得安裝支援。若要利用此服務，請在SUSE LINUX Portal (<http://portal.suse.com>) 以CD盒上印製的代碼啓動您的支援授權。

要確保您的系統一直保持在最新且安全的狀態，建議您使用便利的YaST線上更新進行定期更新。SUSE也提供免費的電子報，內容包括安全相關資訊及SUSE LINUX使用技巧與秘訣。只要在<http://www.novell.com/company/subscribe/> 輸入您的電子郵件地址就可以訂閱。

SUSE LINUX 管理指南提供您SUSE LINUX系統作業方式的背景資訊。本手冊爲您介紹Linux系統管理的基本概念，如檔案系統、核心、開機程序以及Apache網頁伺服器的組態。SUSE LINUX 管理指南內容共可分成五大主題：

安裝 使用YaST進行系統安裝與設定，提供有關特殊安裝類型、LVM、RAID、更新和系統復原等資訊。

系統 SUSE LINUX的特殊功能，提供有關核心、啓動概念和開機程序、開機載入程式與X Window系統的組態、列印，以及Linux中的行動運算功能的詳細資訊。

服務 異質網路的整合、Apache網頁伺服器的組態、檔案同步化，以及安全性等方面。

管理 檔案系統ACL與重要的系統監視工具。

附錄 提供重要的Linux資訊來源。

SUSE LINUX手冊的數位版本位於 `/usr/share/doc/manual/` 目錄。

管理指南中的修訂

舊版本(SUSE LINUX 9.2) 的文件已做的修改如下：

- 有關LVM及磁碟分割的內容已經過改寫。請參閱節3.7, “LVM組態”及節2.7.5, “磁碟分割”。
- 章8, 開機載入器已經過改寫，並加入有關YaST模組的說明。同時也包含新的章節，說明萬用字元的使用方式(節8.3.1, “使用萬用字元以選取開機核心”)。
- 檔案系統章節現在包含了有關Reiser4檔案系統的資訊。請參閱節20.2.5, “Reiser4”。
- 有關網路部份的章節已經重新調整架構並全部改寫。請參閱章22, 基本網路及後續章節。
- SuSEfirewall2已經更新，並加入有關新YaST模組的說明。請參閱節34.1.4, “使用YaST進行設定”。
- 章36, 系統監視公用程式介紹多項新程式。
- 字彙的部份已經經過改寫與更新，請參閱。

印刷慣例

本手冊使用下列印刷慣例：

- `/etc/passwd`：檔案或目錄名稱
- `<保留字元>`：以實際的值來取代`<保留字元>`
- `PATH`：環境變數`PATH`
- `ls`：指令
- `--help`：選項與參數
- 使用者：使用者
- `(Alt)`：要按的鍵

- ‘檔案’：功能表項目、按鈕
- 程序已刪除：系統訊息
- man man(1)：參照man 頁面
- ► x86, AMD64
本章節僅與指定的架構有關。箭頭標示了文字區塊的開頭與結尾。◀

聲明

在許多人自發的努力之下，全球的Linux 開發人員促進了Linux 的發展。我們十分感謝他們的努力，沒有他們，這個版本不會存在。不僅如此，我們還要謝謝Frank Zappa 與Pawar。當然還要特別感謝Linus Torvalds。

祝您使用愉快！

您的SUSE 團隊

Contents

I	安裝	1
1	使用YaST 安裝	3
1.1	系統啓動進行安裝	4
1.1.1	開機選項	4
1.1.2	系統開機時可能發生的問題	4
1.2	開機畫面	5
1.3	語言選擇	7
1.4	安裝模式	7
1.5	安裝建議	7
1.5.1	安裝模式	8
1.5.2	鍵盤配置	8
1.5.3	滑鼠	8
1.5.4	磁碟分割	9
1.5.5	軟體	16
1.5.6	開機組態	19
1.5.7	時區	19
1.5.8	語言	19
1.5.9	啓動安裝	21
1.6	完成安裝	21
1.6.1	根密碼	21

1.6.2	網路組態	22
1.6.3	防火牆組態	22
1.6.4	測試網際網路連線	23
1.6.5	載入軟體更新	23
1.6.6	使用者認證	24
1.6.7	將主機設定為NIS 用戶端	25
1.6.8	建立本地使用者帳戶	26
1.6.9	發行記事	28
1.7	硬體組態	28
1.8	圖形登入	29
2	使用YaST 進行系統組態	31
2.1	YaST 控制中心	32
2.2	軟體	33
2.2.1	安裝和移除軟體	33
2.2.2	變更安裝來源	39
2.2.3	YaST 線上更新	39
2.2.4	修補程式光碟更新	41
2.2.5	系統更新	42
2.2.6	媒體檢查	44
2.3	硬體	44
2.3.1	CD-ROM 與DVD 光碟機	44
2.3.2	印表機	44
2.3.3	硬碟控制器	45
2.3.4	硬體資訊	45
2.3.5	IDE DMA 模式	45
2.3.6	掃描器	46
2.3.7	聲音	48
2.3.8	電視與廣播卡	49
2.4	網路設備	49
2.5	網路服務	50

2.5.1	郵件轉送代辦程式	50
2.5.2	其他可用的服務	50
2.6	保全性與使用者	52
2.6.1	使用者管理	52
2.6.2	群組管理	53
2.6.3	安全性設定	54
2.6.4	防火牆	55
2.7	系統	56
2.7.1	系統區域的備份複本	56
2.7.2	還原系統	57
2.7.3	建立開機磁片與救援磁片	57
2.7.4	LVM	58
2.7.5	磁碟分割	58
2.7.6	設定檔管理員(SCPM)	63
2.7.7	Systems Servcices (RunLevel)	63
2.7.8	Sysconfig 編輯器	63
2.7.9	時區選擇	63
2.7.10	語言選擇	64
2.8	其他	64
2.8.1	提交支援要求	64
2.8.2	開機記錄	64
2.8.3	系統記錄	64
2.8.4	載入廠商的驅動程式光碟	65
2.9	文字模式(ncurses) 中執行YaST	65
2.9.1	在模組中瀏覽	66
2.9.2	組合鍵的限制	68
2.9.3	啓動個別模組	68
2.9.4	YOU 模組	68
2.10	從指令行進行線上更新	69

3	特別安裝程序	71
3.1	linuxrc	72
3.1.1	傳送參數到linuxrc	72
3.2	VNC 的安裝	73
3.2.1	準備安裝VNC	74
3.2.2	要安裝VNC的用戶端	74
3.3	使用YaST 進行以文字為基礎的安裝	74
3.4	啓動SUSE LINUX	75
3.4.1	圖形化SUSE 畫面	76
3.4.2	停用SUSE 畫面	76
3.5	提示和秘訣	77
3.5.1	使用rawwritewin 建立開機片	77
3.5.2	使用rawrite 建立開機片	77
3.5.3	在UNIX 類型系統建立開機片	78
3.5.4	從磁片(SYSLINUX) 開機	79
3.5.5	不支援的CD-ROM 光碟機	80
3.5.6	從網路來源安裝	80
3.6	指定永久性設備檔案名稱至SCSI 設備	80
3.7	LVM 組態	81
3.7.1	邏輯磁碟區管理	81
3.7.2	使用YaST 設定LVM 組態	83
3.8	軟體RAID 組態	86
3.8.1	軟體RAID	86
3.8.2	使用YaST 進行軟體RAID 組態	88
3.8.3	疑難排解	89
3.8.4	若需詳細資訊	90

4	更新系統和套件管理	91
4.1	更新SUSE LINUX	92
4.1.1	準備	92
4.1.2	可能的問題	92
4.1.3	使用YaST 更新	93
4.1.4	更新個別套件	93
4.2	軟體隨著版本而變更	93
4.2.1	從8.1 到8.2	94
4.2.2	從8.2 到9.0	95
4.2.3	從9.0 到9.1	95
4.2.4	從9.1 到9.2	101
4.2.5	從9.2 到9.3	107
4.3	RPM — 套件管理員	109
4.3.1	確認套件驗證性	109
4.3.2	管理套件：安裝、更新與解除安裝	110
4.3.3	RPM 與修補程式	111
4.3.4	Delta RPM 套件	112
4.3.5	RPM 查詢	113
4.3.6	安裝與編譯來源套件	115
4.3.7	以build 編譯RPM 套件	117
4.3.8	RPM 歸檔和RPM 資料庫工具	117
5	系統修復	119
5.1	自動修復	120
5.2	使用者定義的修復	121
5.3	進階工具	122
5.4	SUSE 救援系統	122
5.4.1	啓動救援系統	123
5.4.2	使用救援系統	123

II	系統	125
6	64 位元系統環境的32 位元和64 位元應用程式	127
6.1	執行期間支援	128
6.2	軟體開發	128
6.3	Biarch 平台的軟體編譯	129
6.4	核心規格	130
7	啓動及設定Linux 系統	131
7.1	Linux 開機程序	132
7.1.1	initrd	133
7.1.2	linuxrc	133
7.1.3	若需詳細資訊	134
7.2	init 程式	134
7.3	Runlevel	135
7.4	變更Runlevel	136
7.5	Init 程序檔	137
7.5.1	新增init 程序檔	139
7.6	Systems Servcices (RunLevel)	140
7.7	SuSEconfig 和/etc/sysconfig	141
7.8	YaST sysconfig 編輯器	143
8	開機載入器	145
8.1	開機管理	146
8.2	選取開機載入器	147
8.3	使用GRUB 開機	147
8.3.1	GRUB 開機功能表	148
8.3.2	檔案device.map	153
8.3.3	檔案/etc/grub.conf	153
8.3.4	GRUB 外圍程序	154
8.3.5	設定啓動密碼	154

8.4	使用YaST 設定開機載入程式	155
8.4.1	主視窗	156
8.4.2	開機載入程式組態選項	157
8.5	解除安裝Linux 開機載入器	158
8.6	建立開機CD	158
8.7	圖形化SUSE 畫面	159
8.8	疑難排解	160
8.9	若需詳細資訊	161
9	Linux 核心	163
9.1	核心更新	164
9.2	核心來源	164
9.3	核心組態	164
9.3.1	指令行上的組態	165
9.3.2	文字模式中的組態	165
9.3.3	X Window 系統中的組態	165
9.4	核心模組	166
9.4.1	使用hwinfo 的說明進行硬體偵測	166
9.4.2	處理模組	166
9.4.3	/etc/modprobe.conf	167
9.4.4	Kmod — 核心模組載入器	167
9.5	編譯核心	168
9.6	安裝核心	168
9.7	編譯後清理硬碟	169
10	SUSE LINUX 的特殊功能	171
10.1	特殊軟體套件的資訊	172
10.1.1	Bash 套件與/etc/profile	172
10.1.2	cron 套件	172
10.1.3	記錄檔：logrotate 套件	173
10.1.4	Man 頁面	174

10.1.5	locate 指令	174
10.1.6	ulimit 指令	174
10.1.7	free 指令	175
10.1.8	/etc/resolv.conf 檔案	176
10.1.9	GNU Emacs 的設定	176
10.1.10	vi 的概略介紹	177
10.2	虛擬主控台	179
10.3	鍵盤配置	179
10.4	語言與國家專用的設定	180
10.4.1	一些範例	181
10.4.2	語言支援的設定	182
11	X Window System	183
11.1	使用SaX2 來設定X11	184
11.1.1	桌面	185
11.1.2	圖形介面卡	186
11.1.3	色彩與解析度	187
11.1.4	虛擬解析度	188
11.1.5	3D 加速功能	188
11.1.6	影像位置與大小	189
11.1.7	多重顯示器	189
11.1.8	輸入設備	190
11.1.9	AccessX	191
11.1.10	若需詳細資訊	192
11.1.11	搖桿	192
11.1.12	鍵盤配置選擇	192
11.1.13	滑鼠	192
11.2	X 組態最佳化	192
11.2.1	螢幕段落	194
11.2.2	設備段落	196
11.2.3	顯示器和模式段落	197

11.3	安裝與設定字型	197
11.3.1	Xft	198
11.3.2	X11 核心字型	201
11.3.3	CID-Keyed 字型	201
11.4	OpenGL — 3D 組態	202
11.4.1	硬體支援	202
11.4.2	OpenGL 驅動程式	203
11.4.3	診斷工具3Ddiag	203
11.4.4	OpenGL 測試公用程式	203
11.4.5	疑難排解	203
11.4.6	安裝支援	204
11.4.7	其他線上文件	204
12	印表機操作	205
12.1	準備與其他考量	206
12.2	列印系統的工作流程	207
12.3	連線印表機的方法和通訊協定	207
12.4	安裝軟體	208
12.5	設定印表機	208
12.5.1	本地印表機	208
12.5.2	網路印表機	211
12.5.3	組態工作	212
12.6	應用程式的組態	213
12.6.1	由指令行開始列印	213
12.6.2	使用指令行工具由應用程式列印	214
12.6.3	使用CUPS 列印系統	214
12.7	SUSE LINUX 中的特殊功能	214
12.7.1	CUPS 伺服器與防火牆	214
12.7.2	CUPS 網路前端管理員	215
12.7.3	CUPS 列印服務(cupsd) 中的變更	215
12.7.4	各種套件中的PPD 檔案	217

12.8	疑難排解	218
12.8.1	沒有標準印表機語言模式支援的印表機	219
12.8.2	PostScript 印表機沒有可用的PPD 檔案	219
12.8.3	並列埠	219
12.8.4	網路印表機連線方式	220
12.8.5	列印成品損毀而無錯誤訊息	222
12.8.6	停用佇列	222
12.8.7	CUPS 瀏覽：刪除列印工作	222
12.8.8	損毀的列印工作與資料傳輸錯誤	223
12.8.9	除錯CUPS 列印系統	223
12.8.10	若需詳細資訊	224
13	Linux 的行動計算功能	225
13.1	筆記型電腦	226
13.1.1	筆記型電腦硬體的特殊屬性	226
13.1.2	省電	226
13.1.3	與變動作業環境的整合	227
13.1.4	軟體選項	228
13.1.5	資料安全性	230
13.2	行動硬體	231
13.3	行動電話和PDA	231
13.4	若需詳細資訊	232
14	PCMCIA	233
14.1	硬體	234
14.2	軟體	234
14.2.1	基本模組	234
14.2.2	卡片管理員	234
14.3	組態	235
14.3.1	網路卡	236
14.3.2	ISDN	236

14.3.3	數據機	236
14.3.4	SCSI 與IDE	236
14.4	公用程式	237
14.5	疑難排解	237
14.5.1	PCMCIA 基本系統無法運作	237
14.5.2	PCMCIA 卡無法正常運作	238
14.6	若需詳細資訊	239
15	系統組態設定檔管理	241
15.1	詞彙	242
15.2	使用指令行設定SCPM	242
15.2.1	啓動SCPM 與定義資源群組	243
15.2.2	建立與管理設定檔	243
15.2.3	切換組態設定檔	244
15.2.4	進階設定檔設定	244
15.3	YaST 設定檔管理員	245
15.3.1	設定資源群組	245
15.3.2	建立新設定檔	246
15.3.3	修改現有設定檔	246
15.3.4	切換設定檔	247
15.4	疑難排解	248
15.4.1	在切換過程中終止	249
15.4.2	變更資源群組組態	249
15.5	在系統開機時選取設定檔	249
15.6	若需詳細資訊	249

16	電源管理	251
16.1	省電功能	252
16.2	APM	253
16.3	ACPI	254
16.3.1	ACPI 的動作	254
16.3.2	控制CPU 效能	257
16.3.3	ACPI 工具	257
16.3.4	疑難排解	258
16.4	硬碟的休眠	259
16.5	powersave 套件	260
16.5.1	設定powersave 套件	261
16.5.2	設定APM 和ACPI	262
16.5.3	其他ACPI 功能	264
16.5.4	疑難排解	265
16.6	YaST電源管理模組	267
17	無線通訊	273
17.1	無線區域網路	274
17.1.1	硬體	274
17.1.2	功能	275
17.1.3	使用YaST 進行設定	277
17.1.4	公用程式	279
17.1.5	設定WLAN 的秘訣與技巧	279
17.1.6	疑難排解	280
17.1.7	若需詳細資訊	280
17.2	藍芽	281
17.2.1	基本操作	281
17.2.2	組態	282
17.2.3	系統元件和公用程式	285
17.2.4	圖形應用程式	286
17.2.5	範例	286

17.2.6	疑難排解	287
17.2.7	若需詳細資訊	289
17.3	紅外線資料傳輸	289
17.3.1	軟體	289
17.3.2	組態	289
17.3.3	使用方式	290
17.3.4	疑難排解	290
18	熱插拔系統	293
18.1	設備和介面	294
18.2	熱插拔事件	295
18.3	熱插拔代辦	295
18.3.1	啓動網路介面	296
18.3.2	啓動儲存設備	296
18.4	自動化模組載入	297
18.5	PCI的熱插拔	297
18.6	開機程序檔冷插拔	298
18.7	錯誤分析	298
18.7.1	日誌檔	298
18.7.2	開機問題	298
18.7.3	事件記錄器	298
19	使用udev的動態裝置節點	301
19.1	建立規則	302
19.2	以NAME與SYMLINK自動化	303
19.3	關鍵字中的一般表示式	303
19.4	關鍵字選擇	303
19.5	大容量儲存裝置的一致名稱	304

20 Linux 的檔案系統	307
20.1 詞彙	308
20.2 Linux 的主要檔案系統	308
20.2.1 ReiserFS	309
20.2.2 Ext2	309
20.2.3 Ext3	310
20.2.4 將Ext2 檔案系統轉換成Ext3	311
20.2.5 Reiser4	311
20.2.6 JFS	312
20.2.7 XFS	312
20.3 其他支援的檔案系統	313
20.4 Linux 的大型檔案支援	314
20.5 若需詳細資訊	315
21 使用PAM 驗證	317
21.1 PAM 組態檔的結構	318
21.2 sshd 的PAM 組態	319
21.3 PAM 模組的組態	321
21.3.1 pam_unix2.conf	321
21.3.2 pam_env.conf	322
21.3.3 pam_pwcheck.conf	323
21.3.4 limits.conf	323
21.4 若需詳細資訊	323
III 服務	325
22 基本網路	327
22.1 IP 位址與路由	330
22.1.1 IP 位址	330
22.1.2 網路遮罩與路由	331
22.2 IPv6 — 下一代的網際網路	333

22.2.1	優點	333
22.2.2	定址類型與結構	334
22.2.3	IPv4 與IPv6 的共存	338
22.2.4	設定IPv6	339
22.2.5	若需詳細資訊	339
22.3	名稱解析	339
22.4	使用YaST 手動設定網路連線	340
22.4.1	使用YaST 設定網路卡	341
22.4.2	數據機	343
22.4.3	ISDN	344
22.4.4	纜線數據機	347
22.4.5	DSL	347
22.5	手動設定網路連線	349
22.5.1	組態檔	351
22.5.2	啓動程序檔	357
22.6	smpppd 做爲撥號助理	357
22.6.1	設定smpppd	358
22.6.2	設定kinternet、cinternet 與qinternet 以供遠端使用	359
23	網路中的SLP 服務	361
23.1	註冊自己的服務	362
23.2	在SUSE LINUX 中的SLP 前端	363
23.3	啓動SLP	363
23.4	若需詳細資訊	363
24	領域名稱系統	365
24.1	使用YaST 進行設定	366
24.1.1	精靈組態	366
24.1.2	進階組態	367
24.2	啓動名稱伺服器BIND	370
24.3	組態檔/etc/named.conf	374

24.3.1	重要組態選項	375
24.3.2	記錄	376
24.3.3	區域項目	376
24.4	區域檔案	378
24.5	區域資料的動態更新	381
24.6	安全交易	381
24.7	DNS 安全性	382
24.8	若需詳細資訊	383
25	使用NIS	385
25.1	設定NIS 伺服器	386
25.2	設定NIS 用戶端	388
26	使用NFS 共用檔案系統	391
26.1	以YaST 輸入檔案系統	392
26.2	手動輸入檔案系統	392
26.3	以YaST 輸出檔案系統	393
26.4	手動輸出檔案系統	394
27	DHCP	397
27.1	使用YaST 設定DHCP 伺服器	398
27.2	DHCP 軟體套件	399
27.3	DHCP 伺服器dhcpd	400
27.3.1	使用固定IP 位址的用戶端	403
27.3.2	SUSE LINUX 版本	404
27.4	若需詳細資訊	404
28	使用xntp 進行時間同步化	405
28.1	設定網路中的xntp	406
28.2	設定本機參考時鐘	406
28.3	使用YaST 設定NTP 用戶端	407
28.3.1	快速NTP 用戶端組態	407
28.3.2	複雜的NTP 用戶端組態	408

29 LDAP — 一種目錄服務	411
29.1 LDAP 與NIS 的比較	413
29.2 LDAP 目錄樹的結構	413
29.3 使用slapd.conf 來設定伺服器	416
29.3.1 slapd.conf 中的全域指示詞	416
29.3.2 slapd.conf 中的資料庫特定指示詞	419
29.3.3 啓動及停止伺服器	420
29.4 LDAP 目錄中的資料處理	420
29.4.1 將資料加入LDAP 目錄	420
29.4.2 修改LDAP 目錄中的資料	422
29.4.3 搜尋或讀取LDAP 目錄中的資料	423
29.4.4 刪除LDAP 目錄中的資料	423
29.5 YaST LDAP 用戶端	424
29.5.1 標準程序	424
29.5.2 LDAP 用戶端的組態	425
29.5.3 使用者和群組— 使用YaST 設定組態	429
29.6 若需詳細資訊	429
30 Apache 網頁伺服器	433
30.1 基本操作	434
30.1.1 網頁伺服器	434
30.1.2 HTTP	434
30.1.3 URL	434
30.1.4 自動顯示預設網頁	435
30.2 使用YaST 設定HTTP 伺服器	435
30.3 Apache 模組	435
30.4 執行緒	436
30.5 安裝	437
30.5.1 選取YaST 中的套件	437
30.5.2 啓動Apache	437
30.5.3 主動內容的模組	437

30.5.4	其他建議套件	437
30.5.5	使用apxs 安裝模組	438
30.6	組態	438
30.6.1	使用SuSEconfig 進行設定	438
30.6.2	手動組態	439
30.7	使用Apache	442
30.8	主動內容	443
30.8.1	Server Side Includes	443
30.8.2	通用閘道介面(CGI)	444
30.8.3	GET 和POST	444
30.8.4	使用模組產生主動內容	444
30.8.5	mod_perl	445
30.8.6	mod_php4	447
30.8.7	mod_python	447
30.8.8	mod_ruby	447
30.9	虛擬主機	447
30.9.1	以名稱為基礎的虛擬主機	448
30.9.2	以IP 為基礎的虛擬主機	448
30.9.3	多個Apache 例項	449
30.10	安全性	450
30.10.1	將風險降到最小	450
30.10.2	存取許可權	450
30.10.3	保持更新	451
30.11	疑難排解	451
30.12	若需詳細資訊	451
30.12.1	Apache	451
30.12.2	CGI	452
30.12.3	安全性	452
30.12.4	其他來源	452

31 檔案同步化	453
31.1 可用的資料同步化軟體	454
31.1.1 Unison	454
31.1.2 CVS	454
31.1.3 subversion	455
31.1.4 mailsync	455
31.1.5 rsync	455
31.2 選取程式時所要考慮的決定性因素	455
31.2.1 用戶端對伺服器與點對點	455
31.2.2 可攜式	456
31.2.3 互動式與自動化	456
31.2.4 衝突：事件與解決方案	456
31.2.5 選取和新增檔案	456
31.2.6 歷程記錄	457
31.2.7 資料量與硬碟需求	457
31.2.8 GUI	457
31.2.9 使用者親切性	457
31.2.10 防止攻擊的安全性	457
31.2.11 針對資料遺失的防護	458
31.3 Unison 的介紹	458
31.3.1 需求	459
31.3.2 使用Unison	459
31.3.3 若需詳細資訊	460
31.4 CVS 簡介	460
31.4.1 設定CVS 伺服器	460
31.4.2 使用CVS	461
31.4.3 若需詳細資訊	462
31.5 Subversion 簡介	462
31.5.1 安裝Subversion 伺服器	462
31.5.2 使用與作業	463

31.5.3	若需詳細資訊	465
31.6	rsync 簡介	465
31.6.1	組態與作業	465
31.6.2	若需詳細資訊	466
31.7	mailsync 簡介	467
31.7.1	組態與使用	467
31.7.2	可能的問題	469
31.7.3	若需詳細資訊	469
32	Samba	471
32.1	設定伺服器	473
32.1.1	全域區段	473
32.1.2	共用	474
32.1.3	安全性層級	476
32.2	做為登入伺服器的Samba	476
32.3	使用YaST 設定Samba 伺服器	478
32.4	設定用戶端	479
32.4.1	使用YaST 設定Samba 用戶端	479
32.4.2	Windows 9x 與ME	479
32.5	最佳化	480
33	Proxy 伺服器：Squid	483
33.1	做為Proxy 快取的Squid	483
33.2	關於Proxy 快取的說明	484
33.2.1	Squid 以及安全性	484
33.2.2	多個快取	484
33.2.3	快取網際網路物件	485
33.3	系統需求	485
33.3.1	硬碟	485
33.3.2	磁碟快取的大小	485
33.3.3	RAM	486

33.3.4	CPU	486
33.4	啓動Squid	486
33.4.1	開始和停止Squid的指令	486
33.4.2	本地DNS伺服器	487
33.5	/etc/squid/squid.conf 組態檔案	488
33.5.1	一般組態選項(選擇)	488
33.5.2	存取控制的選項	490
33.6	設定操作順暢的Proxy	492
33.6.1	核心組態	492
33.6.2	在/etc/squid/squid.conf 中的組態選項	493
33.6.3	使用SuSEfirewall2 的防火牆組態	493
33.7	cachemgr.cgi	495
33.7.1	設定	495
33.7.2	在/etc/squid/squid.conf 中的快取管理員ACL	495
33.7.3	檢視統計資料	496
33.8	squidGuard	496
33.9	使用Calamaris 產生快取報告	498
33.10	詳細資訊	498

IV 管理 499

34	Linux 的安全性	501
34.1	偽裝與防火牆	502
34.1.1	使用iptables 過濾封包	502
34.1.2	偽裝基本原則	504
34.1.3	防火牆基本原則	505
34.1.4	SuSEfirewall2	506
34.1.5	若需詳細資訊	510
34.2	SSH：安全網路作業	510
34.2.1	OpenSSH 套件	510

34.2.2	ssh 程式	510
34.2.3	scp — 安全複製	511
34.2.4	sftp — 安全檔案傳輸	511
34.2.5	SSH 精靈(sshd) — 伺服器端	512
34.2.6	SSH 驗證機制	513
34.2.7	X, 驗證與轉寄機制	513
34.3	加密分割區和檔案	514
34.3.1	應用程式案例	514
34.3.2	使用YaST 設定密碼檔案系統	514
34.3.3	加密抽取式媒體內容	516
34.4	安全性與機密性	516
34.4.1	本地安全性與網路安全性	517
34.4.2	一些一般的安全性秘訣與技巧	523
34.4.3	使用集中式安全性報告位址	525
35	Linux 存取控制清單	527
35.1	ACL 的優點	528
35.2	定義	528
35.3	處理ACL	529
35.3.1	ACL 項目和檔案模式許可權位元	530
35.3.2	含存取ACL 的目錄	531
35.3.3	含預設ACL 的目錄	533
35.3.4	ACL 檢查演算法	535
35.4	應用程式的ACL 支援	536
35.5	若需詳細資訊	536

36 系統監視公用程式	537
36.1 開啓檔案的清單：lsdf	538
36.2 使用者存取檔案：fuser	539
36.3 檔案內容：stat	539
36.4 USB 設備：lsusb	540
36.5 SCSI 設備的相關資訊：scsiinfo	541
36.6 處理程序：top	541
36.7 程序清單：ps	542
36.8 程序樹：pstree	543
36.9 功能介紹：w	544
36.10 記憶體的使用：free	545
36.11 核心環狀緩衝區：dmesg	546
36.12 檔案系統及其使用：mount、df 和du	546
36.13 /proc 檔案系統	547
36.14 vmstat、iostat 及mpstat	549
36.15 procinfo	549
36.16 PCI 資源：lspci	550
36.17 執行程式的系統呼叫：strace	551
36.18 一個程式執行階段的程式庫呼叫：ltrace	552
36.19 指定需要的程式庫：ldd	552
36.20 其它有關ELF二進位的資訊	553
36.21 程序間的通訊：ipcs	554
36.22 使用time 進行時間管理	554
V 附錄	555
A 資訊來源和文件	557
B 檔案系統檢查	561
C GNU 通用公共授權(General Public License)	577
詞彙	583

Part I

安 装

使用YaST 安裝

本章透過系統助手YaST，在整個SUSE LINUX 系統安裝過程中提供有系統的指導。安裝程序準備的敘述是透過背景資訊來達成，以便協助您在個別的組態階段中做出正確的決定。

1.1	系統啓動進行安裝	4
1.2	開機畫面	5
1.3	語言選擇	7
1.4	安裝模式	7
1.5	安裝建議	7
1.6	完成安裝	21
1.7	硬體組態	28
1.8	圖形登入	29

1.1 系統啓動進行安裝

請將第一片SUSE LINUX CD或DVD插入光碟機。接著重新啓動電腦，以便從光碟機中的媒體啓動安裝程式。

1.1.1 開機選項

如果從CD或DVD開機發生問題，可以使用其他開機選項。這些選項描述於表格1.1, “開機選項”中。

Table 1.1: 開機選項

開機選項	說明
CD-ROM	這是最簡單的開機選項。如果系統上有Linux支援的本地光碟機，便可以使用這個選項。
磁片	用來產生開機磁片的影像位於CD1的目錄/boot/中。同一目錄中還有一個README。
PXE或BOOTP	只有在系統的BIOS或韌體支援而且網路上有可用的開機伺服器的情況，才支援這個選項。這個作業還可以由其他SUSE LINUX系統處理。
硬碟	SUSE LINUX也可以從硬碟開機。要這樣做，請從第一張光碟的/boot/loader目錄複製核心(linux)和安裝系統(initrd)到硬碟，並在開機載入程式中新增適當的項目。

1.1.2 系統開機時可能發生的問題

如果您使用較舊或不被支援的硬體，從CD或DVD開機便可能發生問題。您的光碟機可能無法讀取CD1上的開機影像。如果是這種情況，請使用CD2來執行系統開機。CD2含有傳統2.88MB的開機影像，即使未支援的磁碟機也可以讀取，並且可以透過網路執行安裝。

BIOS中的開機順序可能設定不正確。請參閱主機板說明文件來取得如何變更BIOS設定的指示。基本指示會在下列段落提供。

BIOS是提供電腦最基本功能的軟體。主機板供應商會針對自己的硬體提供特製的BIOS。通常，BIOS設定只可在特定時間(機器開機時)進行存取。在這個初始

化階段，機器會執行一些硬體診斷測試。其中之一是記憶體檢查，由記憶體計數器指示。當計數器出現時，請尋找指示按下按鍵來存取BIOS設定的一行文字，通常在計數器下方或底端某個位置。該按鍵通常是按(Delete)、(F1)或(Escape)。請按住這個按鍵，直到BIOS設定畫面出現為止。

BIOS 中的鍵盤配置

BIOS 組態通常使用美國鍵盤配置。

要變更AWARD BIOS中的開機順序，請尋找‘BIOS FEATURES SETUP’ (BIOS功能設定) 項目。其他製造商可能使用不同的名稱，例如‘ADVANCED CMOS SETUP’ (進階CMOS設定)。當您找到該項目後，請選取並按(Enter) 確認。

在接著開啓的畫面，請尋找叫做‘BOOT SEQUENCE’ (開機順序) 的子項目。開機順序通常設成C, A 或A, C。在前一種情況，機器首先搜尋硬碟(C)，然後是軟碟機(A) 來尋找可開機媒體。請按(PgUp) 或(PgDown) 鍵來變更設定，直到順序改成A, CDROM, C 為止。

請按(Escape) 來離開BIOS設定畫面。要儲存變更，請選取‘SAVE & EXIT SETUP’ (儲存並結束變更)，也可以按(F10)。要確認儲存設定，請按(Y)。

如果您有SCSI光碟機，請變更SCSI BIOS的設定。例如，在Adaptec主機介面卡，請按(Ctrl)-A 來開啓設定。接著選取‘Disk Utilities’ (磁碟公用程式)，來顯示已經連接的硬體元件。請記下您的光碟機的SCSI ID。請按(Escape) 結束功能表，接著開啓‘Configure Adapter Settings’ (設定介面卡設定)。在‘Additional Options’ (其他選項) 下，請選取‘Boot Device Options’ (開機設備選項)，然後按(Enter)。請輸入光碟機的ID，接著再按(Enter)。接著請按兩下(Escape) 回到SCSI BIOS的開始畫面。退出這個畫面，接著按‘Yes’ (是) 來啓動電腦。

1.2 開機畫面

開機畫面顯示一些安裝程序的選項。‘從硬碟開機’可以啓動已安裝的系統。這是預設的選項，因為磁碟機中經常有留下的光碟。要安裝系統，請以方向鍵選取一個安裝選項。適用的選項有：

安裝 正常安裝模式。將啓用所有先進的硬體功能。

安裝—關閉ACPI 如果正常安裝失敗，可能是因為系統硬體不支援ACPI (進階組態及電源介面)。如果是這種情況，請使用這個選項安裝但不要ACPI支援。

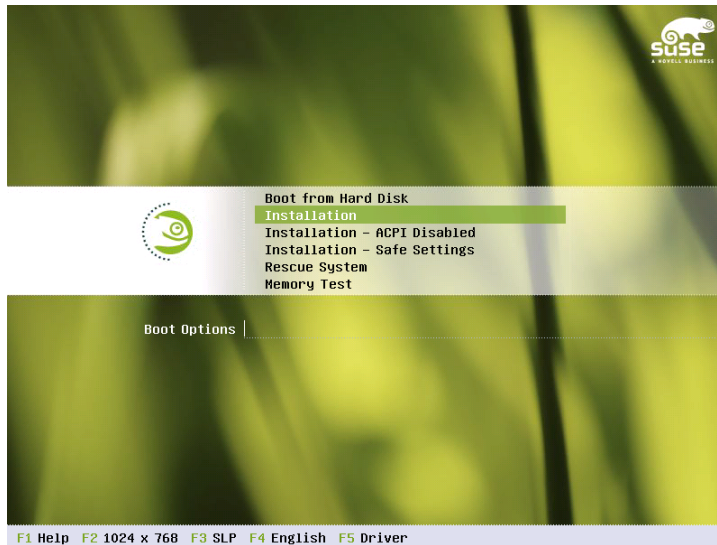


Figure 1.1: 開機畫面

安裝—安全設定 使用DMA 模式(用於光碟機) 開機並關閉電源管理的功能。進階使用者也可以使用指令列來輸入或變更核心參數。

請使用畫面底端列上指示的功能鍵來變更一些安裝設定。

- ⓕ1 提供開機畫面中使用中元素的内容感應式說明。
- ⓕ2 提供進行安裝的各種圖形顯示模式的選擇。如果圖形化安裝會造成問題，可以選擇文字模式。
- ⓕ3 通常，安裝是從插入的安裝媒體執行。也可以從這裡選擇其他來源，例如FTP 或NFS 伺服器。如果安裝是從具有SLP 伺服器的網路執行，可以用這個選項從伺服器上選取一個可用的安裝來源。如需SLP 的相關資訊，請參閱章23, 網路中的SLP 服務。
- ⓕ4 選取安裝的顯示語言。
- ⓕ5 依預設，系統啓動期間不會顯示Linux 核心的診斷訊息。您僅會看到進度列。要顯示這些訊息，請選取‘詳細訊息’。要取得詳細資訊而不要其他的圖形化框架，請選取‘Native’。

- Ⓣ6 這個按鍵可以告訴系統您備有選用光碟，其中含有SUSE LINUX的驅動程式更新。安裝過程中會在適當時機要求您插入該更新光碟。

開始安裝數秒後，SUSE LINUX會載入最小Linux系統，以便執行安裝程序。如果您啓用‘原始’或‘詳細訊息’，會有捲動顯示的訊息以及版權注意事項，載入程序結束後，YaST安裝程式接著啓動。再過數秒鐘，畫面上應該顯示圖形安裝程式。

SUSE LINUX的實際安裝，現在才開始。所有YaST畫面都有共通的外觀。所有按鈕、輸入欄位以及清單，都可以透過滑鼠或鍵盤存取。如果您的滑鼠指標不會移動，表示滑鼠未被自動偵測到。在這種情況，請暫時先使用鍵盤。使用鍵盤進行瀏覽和節2.9.1, “在模組中瀏覽”中的說明相似。

1.3 語言選擇

在一般情況下，您可以依需求將YaST和SUSE LINUX設成使用不同語言。此處選取的語言也會用於鍵盤配置。另外，YaST將使用語言設定來猜測系統時鐘的時區。這些設定可於稍後在選擇將次要語言安裝至系統時進行修改。如果您的滑鼠無法使用，請使用方向鍵選取語言，接著連續按(Tab)，直到‘接受’反白顯示為止。接著按(Enter)來確認您的語言選擇。

1.4 安裝模式

請選取‘全新安裝’或‘更新現有系統’。只有在已經安裝SUSE LINUX系統的情況才可以進行更新。如果是這種情況，已安裝的系統可以用‘啓動已安裝的系統’開機。如果已安裝系統開機失敗，則可能是某個重要的系統組態已經損壞，您可以使用‘修復已安裝的系統’來嘗試系統重新開機。如果沒有已安裝的SUSE LINUX系統，您只能執行全新安裝。請參閱圖形1.3, “選取安裝模式”。

以下數節描述安裝全新系統的程序。如需系統更新的詳細指示，請參閱節2.2.5, “系統更新”。如需系統修復選項的說明，請參閱章5, 系統修復。

1.5 安裝建議

硬體偵測結束後，會出現如圖圖形1.4, “建議視窗”所示的建議視窗，其中顯示一些已識別的硬體資訊以及數個安裝與磁碟分割的選項建議。您選取任一個項目並在對應的對話方塊完成設定後，都將回到內容已經相對更新的建議視窗。以下數節討論這些個別的設定。

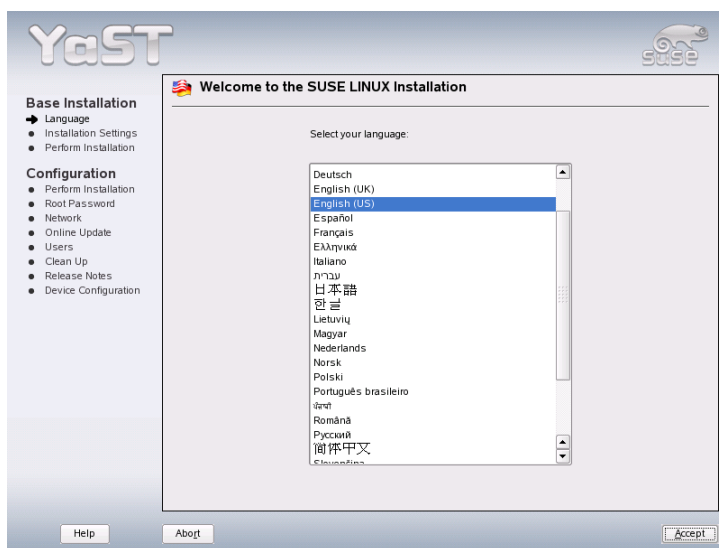


Figure 1.2: 選取語言

1.5.1 安裝模式

使用這個選項來變更之前選取的安裝模式。這個選項和節1.4, “安裝模式” 中描述的相同。

1.5.2 鍵盤配置

選取鍵盤配置。依預設，配置會對應於所選語言。變更配置後，請測試(Y)、(Z) 以及一些特殊字元來確定選擇正確。完成後，請選取‘下一步’ 回到建議視窗。

1.5.3 滑鼠

如果YaST 無法自動偵測滑鼠，請在建議視窗中連續按(Tab)，直到‘滑鼠’ 為已選取狀態為止。接著使用(Space) 來開啓設定滑鼠類型的對話方塊。在圖形1.5, “選取滑鼠類型” 中顯示了此對話方塊。

要選取滑鼠類型，請使用(?) 和(?)。請參閱滑鼠說明文件來取得滑鼠類型的相關資訊。選好滑鼠類型後，請使用(Alt)-(1) 先測試設備是否可以正確運作，而非確實

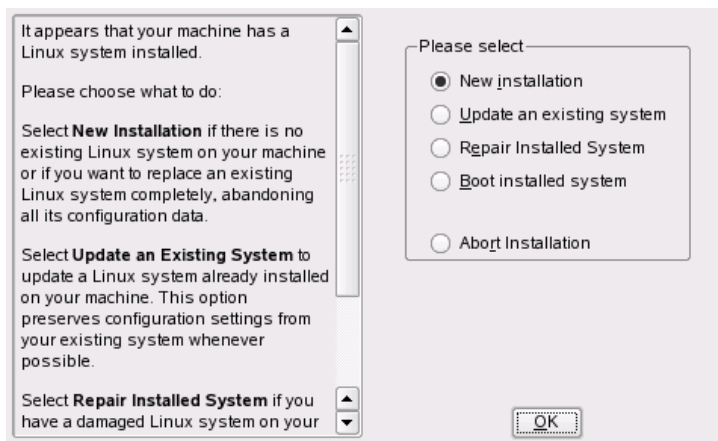


Figure 1.3: 選取安裝模式

選定。如果滑鼠未依預期運作，請使用鍵盤選取其他類型並測試。請使用`(Tab)`和`(Enter)`永久選定目前的選擇。

1.5.4 磁碟分割

在大部份情況，YaST 會建議無需變更便可接受的合理磁碟分割架構。YaST 也可以用來自訂磁碟分割。本節描述必要的步驟。

分割區類型

每一個硬碟都有一個分割區表，可以儲存四筆登錄。分割區表中的一筆登錄可以對應一個主分割區或延伸分割區。不過，延伸分割區登錄只能有一筆。

主分割區僅由指派給特定作業系統之連續範圍的磁柱(實體磁碟區)組成。如果只有主分割區，每一個硬碟將受限於四個分割區，因為無法再容納分割區表。這就是使用延伸分割區的原因。延伸分割區也是由連續範圍的磁柱組成，但延伸分割區又可以再劃分為邏輯分割區。邏輯分割區不需要在分割區表中登錄。換句話說，延伸分割區是邏輯分割區的容器。

如果您需要四個以上的分割區，請建立延伸分割區當作第二至第四個分割區。這個延伸分割區應該包含所有剩餘可用的整個磁柱範圍。接著在延伸分割區中建立多個邏輯分割區即可。SCSI、SATA 以及 Firewire 磁碟的邏輯分割區數上限是15

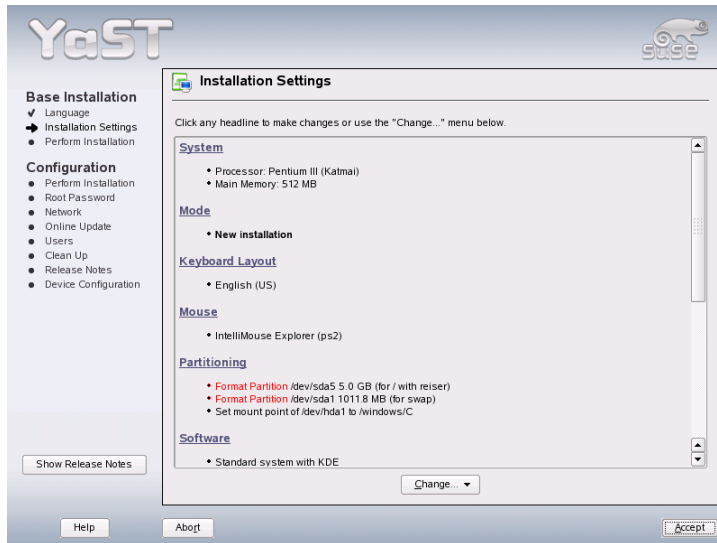


Figure 1.4: 建議視窗

個，(E)IDE 磁碟是63 個。Linux 對於使用的分割區類型沒有限制。主分割區與邏輯分割區都可以正常運作。

具有GPT 磁碟標籤的硬碟

對使用GPT 磁碟標籤的架構來說，並不會限制主分割區數量。因此，在此例中沒有邏輯分割區。

必要的磁碟空間

YaST 通常建議具有足夠磁碟空間的合理磁碟分割架構。如果您想自行實作磁碟分割架構，請考量下列不同系統類型的需求建議。

最小系統：500 MB 不安裝圖形介面(X Window System)，表示只可使用主控台應用程式。此外，將只安裝最基本的軟體選擇。

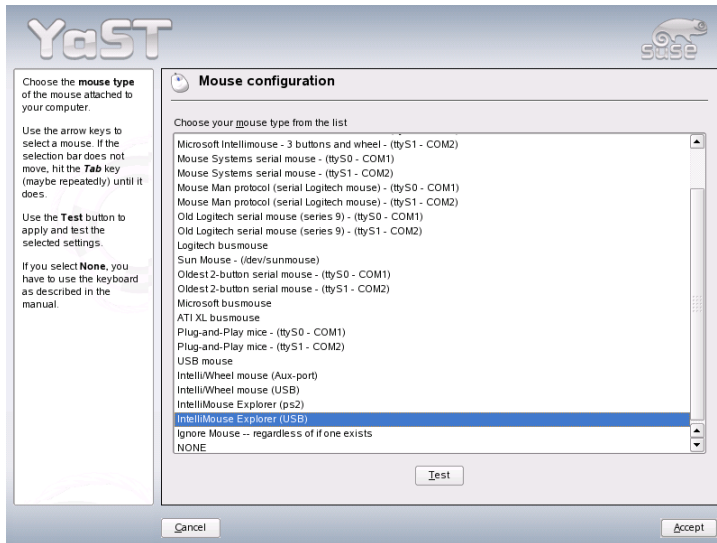


Figure 1.5: 選取滑鼠類型

含圖形介面的最小系統：700 MB 包括X Window System 和一些應用程式。

預設系統：2.5 GB 包括先進的桌面環境如KDE 或GNOME，同時提供足夠的空間安裝大型應用程式套裝軟體如OpenOffice.org 以及Netscape 或Mozilla。

建立的分割區由可用空間決定。以下是基本的磁碟分割指引：

最多4 GB： 一個分割區用於交換空間以及根分割區/。在這種情況時，對於那些在空間足夠的情況下，通常會擁有專用分割區的目錄，根分割區的空間必須足以容納它們。

4 GB 或更多： 一個交換分割區、一個根分割區(1 GB)、以及下列每一個目錄一個分割區(視需要)：/usr (4 GB 或更多)、/opt (4 GB 或更多)、以及/var (1 GB)。如果您不想對這些目錄指定個別的分割區，請將建議的磁碟空間加到根分割區。剩餘的可用空間可以分配給/home。

視硬體而定，也可以建立一個開機分割區(/boot) 來存放開機機制以及Linux 核心，更方便使用。這個分割區應該位於磁碟的開頭而且必須至少8 MB 或一個磁

柱大小。有一個基本原則可供參考，如果YaST已經將它加在原始建議中，請務必建立一個這種分割區。如果您不確定，請建立一個開機分割區以確保安全。

您應該會發現有些(大部份是商用)程式將資料安裝在/opt。因此，請建立個別的/opt分割區，或設定夠大的根分割區。KDE和GNOME也會安裝在/opt。

使用YaST分割磁碟

當您第一次在建議視窗選取磁碟分割項目時，YaST磁碟分割對話方塊會顯示分割區設定當作目前的建議。請接受這些目前的設定或進行變更再繼續。另外，您也可以捨棄這些設定重頭開始。

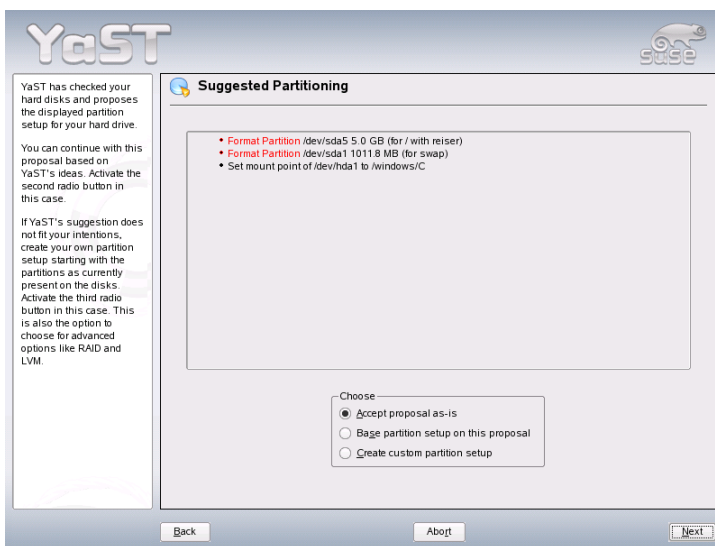


Figure 1.6: 編輯磁碟分割設定

如果您選取‘直接接受建議’，磁碟分割設定不會發生任何變更。如果您選取‘依據這個建議進行分割區設定’，‘進階磁碟分割程式’就會開啓。它容許您進行非常詳細的分割區設定。這個對話方塊的說明在節2.7.5, “磁碟分割”。其中含有YaST建議的原始設定，可以當作起點。

選取‘建立自訂分割區設定’將開啓如圖形1.7, “選取硬碟”所示的對話方塊。請使用其中的清單來選取系統現有的硬碟。SUSE LINUX將安裝到您在這個對話方塊選取的磁碟上。

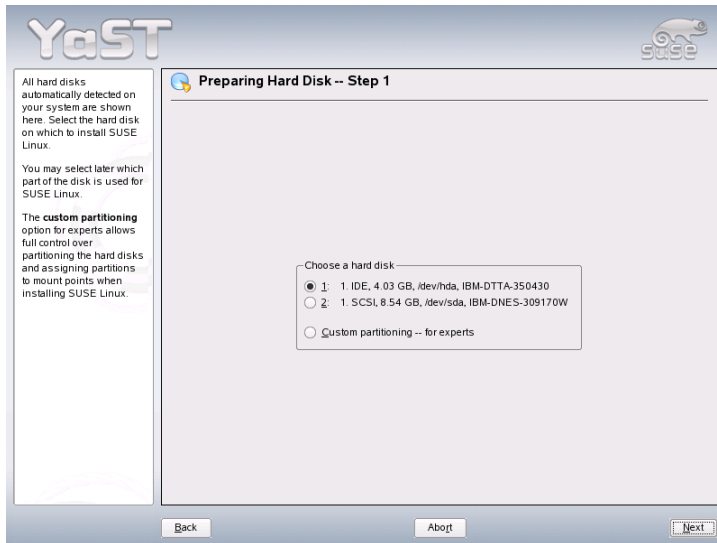


Figure 1.7: 選取 硬 碟

下一個步驟是決定是否使用整個磁碟(‘使用整個硬碟’)或某個現有分割區(如果有的話)來執行安裝。如果磁碟上有Windows作業系統，將會詢問您是否刪除該分割區或調整大小。請先閱讀節1.5.4, “調整Windows分割區的大小”再決定。如果需要，請現在移至‘進階磁碟分割程式’對話方塊來建立自訂的分割區設定(請參閱節2.7.5, “磁碟分割”)。



使用整個硬碟安裝

如果您選擇‘使用整個硬碟’，磁碟上所有現有的資料將在稍後安裝過程中完全移除而遺失。



YaST 會在安裝期間檢查是否有足夠的空間容納選取的軟體。如果沒有，YaST 會自動變更軟體選擇。建議視窗將顯示相關通知。如果可用磁碟空間足夠，YaST 將直接接受您的設定並執行硬碟分割。

調整Windows 分割區的大小

如果您選取含有FAT 或NTFS 分割區的硬碟當作安裝目標，YaST 建議您刪除或縮小該分割區。利用這個方法，即使硬碟上沒有足夠的空間，您仍可以安裝SUSE LINUX。如果您選取的硬碟只含有一個Windows 分割區而且佔用整個硬碟，這個功能特別有用。這是已經預先安裝Windows 的電腦的常見情況。如果YaST 發現所選硬碟上沒有足夠的空間但仍可以刪除或縮小Windows 分割區來取得空間，將會顯示對話方塊供您選擇其中一個選項。

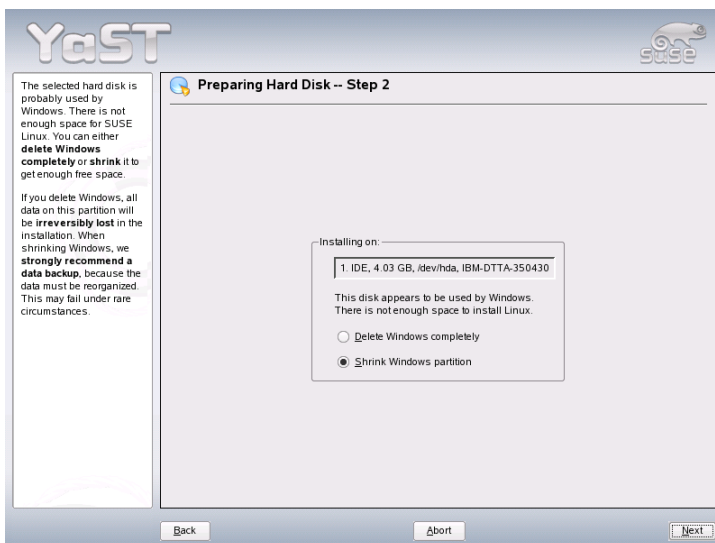


Figure 1.8: Windows 分割區可用的選項

如果您選取‘刪除整個Windows’，Windows 分割區將被標示為刪除，並使用該空間來安裝SUSE LINUX。



刪除Windows

如果您刪除Windows，在格式化作業開始後所有資料都將遺失而且無法復原。



要縮小Windows 分割區，請中斷安裝並啟動Windows 來開始準備分割區。儘管

這個步驟對FAT 分割區並非必要，但可以提高調整大小處理的速度而且更安全。這些步驟對NTFS 分割區十分重要。

FAT 檔案系統 在Windows，請先執行scandisk (磁碟掃描工具) 來確認FAT 分割區沒有遺失的檔案片段以及交叉連結。然後，請執行defrag (磁碟重組工具) 將檔案移至分割區的開頭。這樣做可加速Linux 下的調整大小程序。

如果您已經對Windows 設定最佳化虛擬記憶體設定因而有使用相同的起始(最小) 與最大大小限制的連續交換檔，請考慮使用其他步驟。因為這些Windows 設定，調整大小可能導致交換檔分割成許多小塊並散置在整個FAT 分割區。而且，由於調整大小期間需要移動整個交換檔，會使程序速度更慢。因此，最好的做法是先停用Windows 最佳化設定，並在調整大小完成後再重新啓用。

NTFS 檔案系統 在Windows，請執行scandisk (磁碟掃描工具) 和defrag (磁碟重組工具)，將檔案移至硬碟的開頭。和FAT 檔案系統相反，您必須執行這些步驟。否則無法調整NTFS 分割區的大小。

停用Windows 交換檔

如果您在NTFS 檔案系統上使用永久交換檔來執行系統作業，這個檔案將位於硬碟的尾端，即使使用defrag (磁碟重組工具)，也不會改變位置。因此，縮小的分割區可能不足夠。在這種情況，請暫時關閉交換檔(Windows 中的虛擬記憶體)。調整好分割區大小後，再重新設定虛擬記憶體。

完成這些準備作業後，請回到Linux 磁碟分割設定並選取‘縮小Windows 分割區’。在完成快速檢查分割區後，YaST 會開啓一個對話方塊，提供調整Windows 分割區大小的建議。

第一個長條圖顯示Windows 目前佔用的磁碟空間以及剩餘的可用空間。第二個長條圖顯示依據YaST 目前建議調整大小後得到的空間。請參閱圖形1.9, “調整Windows 分割區大小”。請接受建議的設定值，或者使用滑桿來變更分割區大小(在某些限制內)。

如果您選取‘下一步’ 離開這個對話方塊，將儲存設定值並回到前一個對話方塊。實際調整大小的作業將在稍後磁碟格式化之前開始。

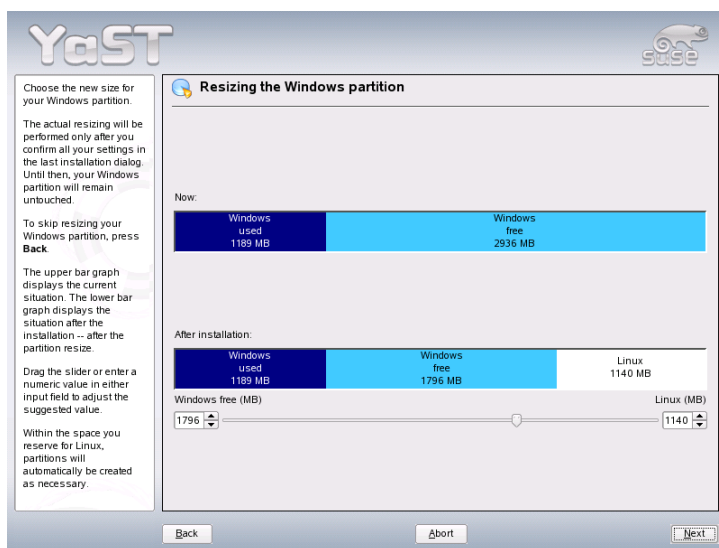


Figure 1.9: 調整Windows 分割區大小

安裝在NTFS 分割區的Windows 系統

依預設，NT、2000 以及XP 等Windows 版本都使用NTFS 檔案系統。不同於FAT 檔案系統，Linux 只可讀取NTFS 檔案系統。這表示您可以從Linux 讀取Windows 檔案，但無法編輯。如果您想寫入存取Windows 資料而且不需要NTFS 檔案系統，請以FAT32 檔案系統重新安裝Windows。在這種情況，您可以從SUSE LINUX 完整存取Windows 資料。

1.5.5 軟體

SUSE LINUX 含有許多滿足各種應用程式目的所需的軟體套件。由於逐一選擇所需套件非常繁瑣，因此SUSE LINUX 提供三種含有不同安裝範圍的系統類

型。YaST 將依據可用的磁碟空間，從中選取一個預先定義的系統並顯示在建議視窗中。

最小系統(僅建議用於特殊目的) 包括基本的核心作業系統以及各種服務，但沒有任何圖形化使用者介面。機器只可透過ASCII主控台操作。這個系統類型特別適用於不需要太多使用者直接介入的伺服器方案。

最小圖形系統(不含GNOME或KDE) 如果您不要KDE或GNOME桌面或沒有足夠的磁碟空間，請安裝這個系統類型。安裝的系統包括X Window System以及一個基本的視窗管理程式。您可以使用具有圖形化使用者介面的所有程式。不會安裝辦公室軟體。

預設系統(含GNOME和辦公室套裝軟體)

這是最大的預先定義系統中的一個。含有GNOME桌面以及大部份GNOME程式和辦公室軟體。

預設系統(含KDE與辦公室套裝軟體) 這個系統包括KDE桌面以及大部份的KDE程式與辦公室程式。

請按一下建議視窗中的‘軟體’來開啓選取預先定義系統的對話方塊。要啓動軟體安裝模組(套件管理員)以及修改安裝範圍，請按一下‘詳細選擇’。請參閱圖形1.10, “使用YaST 套件管理員安裝與移除軟體”。

變更安裝範圍

如果您安裝預設系統，通常不需要新增或移除個別的套件。它是由一組符合大部份需求而且不需要變更的軟體選擇組成。如果您有特殊需求，請使用套件管理員來修改選擇，以簡化作業。它提供多種篩選準則，可以簡化SUSE LINUX中大量套件的選擇。

篩選條件選擇方塊位於功能表列下方左上角。啓動後，使用中的篩選條件在‘選擇’。這個篩選條件依應用程式用途排序程式套件，例如，多媒體或辦公室應用程式。這些群組列出在篩選條件選擇方塊下方。包括在目前系統類型的套件已預先選取。請按一下對應的核取方塊來選取或取消選取安裝整個選擇或群組。

視窗右邊部份顯示一個表格，列出目前選擇的套件中含有的個別套件。表格最左邊的欄位顯示每一個套件的目前狀態。有兩個狀態旗號和安裝關係密切：‘安裝’(套件名稱前的方塊爲已核取)和‘不要安裝’(方塊爲空)。要選取或取消選取個別的軟體套件，請連續按一下狀態方塊，直到所要的狀態顯示出來爲止。此外，也可以在套件行上按一下滑鼠右鍵來存取列出所有可用狀態設定的快顯功能表。如需詳細資訊，請閱讀節2.2.1, “安裝和移除軟體”中這個模組的詳細說明。

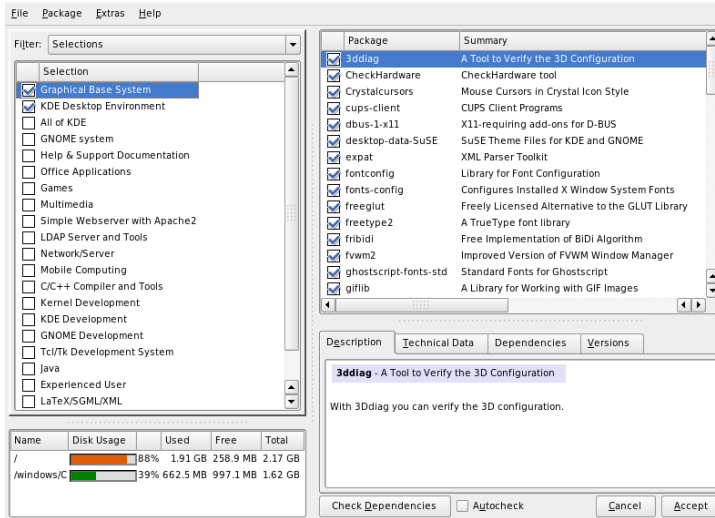


Figure 1.10: 使用YaST 套件管理員安裝與移除軟體

其他篩選條件

按一下篩選條件選擇方塊來檢視其他可用的篩選條件。依據‘套件群組’進行的選擇也可以用於安裝。這個篩選條件會依據左邊樹狀結構的主題來排序程式套件。您展開的分支越多，套件的選擇範圍越小，顯示在右邊關聯套件清單中的套件也越少。

請使用‘搜尋’來尋找特定的套件。如需詳細說明，請參閱節2.2.1, “安裝和移除軟體”。

套件相依性與衝突

您不可以隨意安裝任意軟體套件的組合。各種軟體套件必須相容。否則，它們可能互相干擾並造成衝突而影響系統整體。因此，您在這個對話方塊中選取或取消選取軟體套件後，可能會看到有未解決的套件相依性或衝突的警示。如果您是第一次安裝SUSE LINUX 或者如果您不瞭解警示，請參閱節2.2.1, “安裝和移除軟體”，來取得套件管理員的詳細操作資訊以及Linux 中的軟體組織摘要。

┌

預先選取供安裝的軟體是依據長年累積的經驗，通常最符合新使用者以及進階家庭使用者的需求。通常，這裡不需要進行任何變更。然而，如果您決定選取或取消選取任何套件，請注意其結果。請特別注意是否出現任何警示，同時避免取消選取基礎系統的任何套件。

└

退出軟體選擇

您滿意所做的軟體選擇並解決所有套件相依性或衝突問題後，請按一下‘接受’來套用變更並退出本模組。在安裝期間，變更將先記錄在內部，並在稍後實際安裝啓動時才套用。

1.5.6 開機組態

YaST 將在安裝期間建議您系統的開機組態。通常，您不需要變更這些設定。不過，如果您需要自訂的設定，請修改針對您的系統提供的建議。

您可以將開機機制設成依賴特定開機磁片。雖然這樣表示開機時磁碟機中必須有開機磁片，但是現有的開機機制可以因此不受影響。不過，通常這不是必要的程序，因為YaST 也可以將開機載入程式設成以現有的其他作業系統進行開機。組態的另一個用途是改變硬碟上的開機機制的位置。

要變更YaST 提供的開機組態建議，請選取‘開機’來開啓用來變更多種開機機制的對話方塊。如需更多資訊，請參閱節8.4, “使用YaST 設定開機載入程式”。開機方法的變更應該由經驗豐富的電腦使用者執行。

1.5.7 時區

在這個對話方塊中，如圖形1.11, “選取時區” 所示，您可以在‘硬體時鐘設為’ 選擇當地時間與UTC。選擇由您機器上的硬體(BIOS) 時鐘的設定決定。如果已經設成GMT, 表示和UTC 對應，則您的系統可由SUSE LINUX 自動切換標準時間和日光節約時間。

1.5.8 語言

如節1.3, “語言選擇” 中描述，語言在開始安裝時便已經選擇。不過，您可以在此處變更此設定，也可以選擇將任何其他語言安裝至系統上。在此對話方塊的上

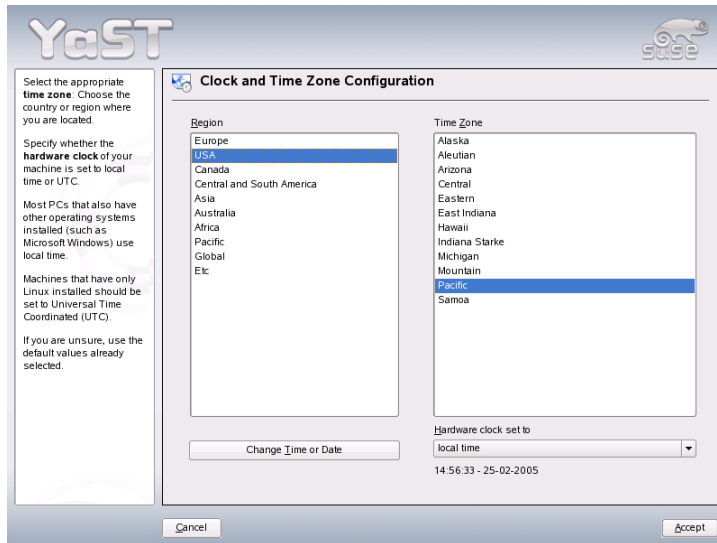


Figure 1.11: 選取時區

半部，選擇主要語言。這個語言會在安裝完成後啓動。若有需要，請選擇對應的核取記號，來調整您的鍵盤配置與時區設定。另外，請使用‘詳細資料’來設定root (根) 使用者的語言。共有三個選項：

僅ctype /etc/sysconfig/language 檔中變數LC_CTYPE 的值將套用於root (根) 使用者。這個值設定語言相關函數的本土化。

yes root (根) 使用者的語言設定和本地使用者相同。

no root (根) 使用者的語言設定不會受到語言選擇影響。所有的地區設定變數均會取消設定。

有些系統管理員不希望root 帳戶執行UTF-8 多語系支援。如果是這樣，請取消選取‘使用UTF-8 編碼’。

在對話方塊的下方清單可以選擇要安裝的其他語言。針對在此清單中所選的所有語言，YaST 會檢查在您目前選擇的軟體中是否有其任何語言相關的套件。如果有，便會安裝這些套件。

請按一下‘接受’以完成設定。請按一下‘取消’來復原變更。

1.5.9 啓動安裝

完成所有安裝設定後，請按一下建議視窗中的‘下一步’開始安裝。請在開啓的對話方塊中選取‘是’進行確認。視系統效能與所選軟體而定，安裝通常需要15至30分鐘。所有套件安裝完成後，YaST將開機進入新的Linux系統，您接著可以設定硬體和系統服務。

1.6 完成安裝

在完成基本的系統設定以及所有選取軟體套件的安裝後，請提供系統管理員(root(根)使用者)帳戶的密碼。您接著可以進行網際網路存取與網路連線的組態。透過實際可用的網際網路連線，您可以執行系統更新並當作安裝的一部份。您還可以設定驗證伺服器，用於集中管理本地網路中的使用者。最後，請設定與機器連接的硬體設備組態。

1.6.1 根密碼

root(根)是超級使用者，系統管理員的名稱。不同於一般使用者，只有對系統進行某些作業的權限，root(根使用者)具有不受限制，可執行任何作業的權限：變更系統組態、安裝程式和設定新硬體。如果使用者忘記密碼或遇到其他系統問題，root(根使用者)可以提供協助。root(根)帳戶應該僅用於系統的管理、維護和修復。以root(根)身份登入來執行日常作業會有很大的風險：一個失誤可能導致遺失許多系統檔案。

如圖形1.12, “設定根密碼”所示，root(根)密碼必須輸入兩次以進行確認。請不要忘記root(根)密碼。一旦輸入後，便無法查詢這個密碼。

┌

根使用者

root(根)使用者具有進行系統變更所需的所有權限。要執行這種作業，必須使用root(根)密碼。您必須透過這個密碼才可以執行任何管理作業。

└

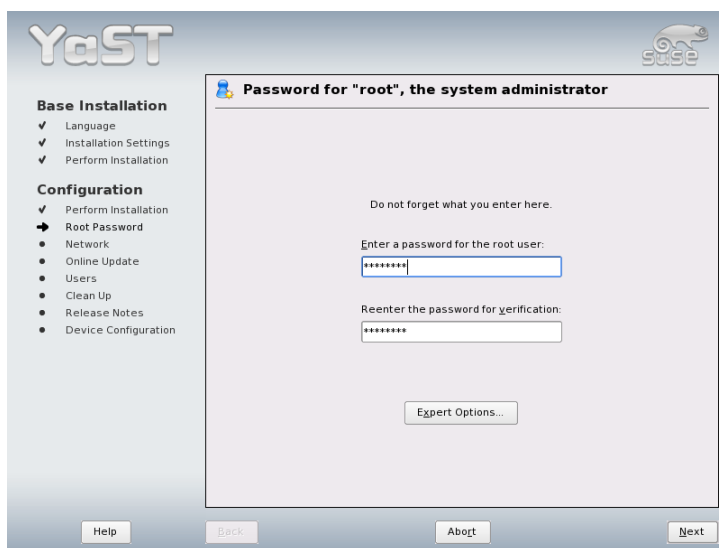


Figure 1.12: 設定根密碼

1.6.2 網路組態

您現在可以設定任何用來連接外部的網路設備，例如，網路卡、數據機和ISDN或DSL硬體。如果您有這些設備，您現在可以進行設定，因為網際網路連線可以讓YaST取得任何可用的SUSE LINUX更新並加到安裝中。

要在這個階段設定網路硬體，請參閱節22.4, "使用YaST手動設定網路連線"。否則，請選取'略過組態'然後按一下'下一步'。您也可以系統安裝完成後，再設定網路硬體。

1.6.3 防火牆組態

當您連上網路時，防火牆會在設定的界面上自動啟動。防火牆設定會顯示在網路組態對話方塊中。防火牆的組態建議會隨每次修改介面或服務的組態自動更新。要將自動設定值加到您的偏好設定，請按一下'變更'→'防火牆'。在新對話方塊，請決定是否要啟動防火牆。如果您不要啟動防火牆，請選取適當的選項並退出對話方塊。要啟動和設定防火牆，請在一系列的對話方塊中按一下'下一步'，如節34.1.4, "使用YaST進行設定"中的敘述。

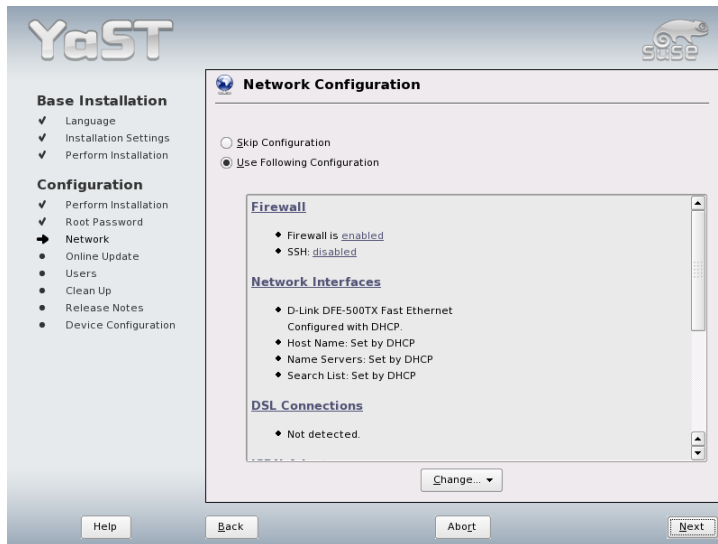


Figure 1.13: 設定網路設備

1.6.4 測試網際網路連線

如果您已經設定網際網路連線，您現在可以進行測試。針對這個目的，YaST 會建立SUSE 伺服器連線來檢查是否有您使用的SUSE LINUX 版本適用的產品更新。如果有適用的更新，將被加到安裝中。另外，最新版的發行記事會一起下載。您會在安裝過程的最後看到。

如果您現在不要測試連線，請選取‘略過測試’，接著按一下‘下一步’。這樣將同時略過產品更新與發行記事的下載。

1.6.5 載入軟體更新

如果YaST 可以連上SUSE 伺服器，請選取是否執行YaST 線上更新。如果伺服器上有適用的修補套件，請下載並安裝以更正已知問題或安全問題。

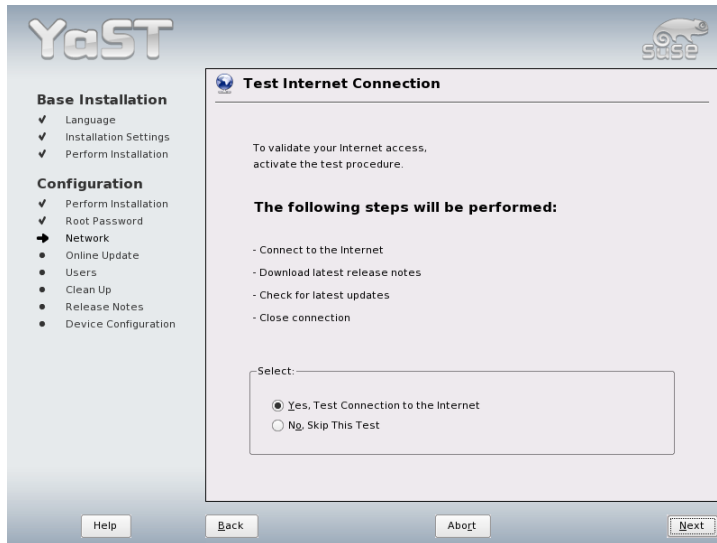


Figure 1.14: 測試網際網路連線

下載軟體更新

下載軟體可能需要一些時間，視網際網路連線的頻寬和更新檔案的大小而定。

要立即執行軟體更新，請選取‘立即執行更新’，接著按一下‘確定’。YaST 的線上更新對話方塊將會開啓，其中顯示可用的修補程式清單(如果有的話)，供您選取與載入。如需這個程序的相關資訊，請參閱節2.2.3, “YaST 線上更新”。安裝後可隨時執行本類型的更新。如果您不要現在更新，請選取‘略過更新’，接著按一下‘確定’。

1.6.6 使用者認證

如果網路存取已經在前面的安裝步驟設定成功，您有四種可用方法管理系統上的使用者帳戶。

本地使用者管理 在安裝主機上本地管理使用者。這是單機工作站適用的選項。使用者資料由本地檔案/etc/passwd 管理。

LDAP 在LDAP 伺服器上集中管理網路中所有系統的使用者。

NIS 在NIS 伺服器上集中管理網路上所有系統的使用者。

Samba Linux 與Windows 混合網路中常使用的SMB 驗證。

如果所有要求都符合，YaST 將開啓一個對話方塊，用來選取使用者管理方法。如圖形1.15, "使用者認證"所示。如果您沒有必要的網路連線，請建立本地使用者帳戶。

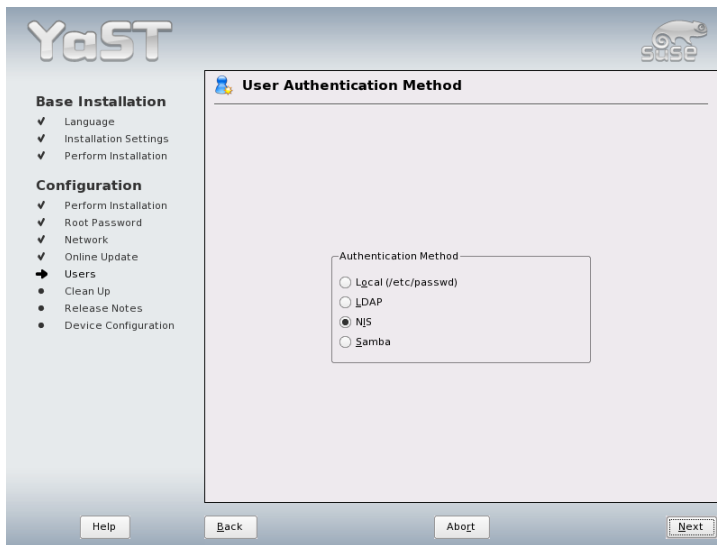


Figure 1.15: 使用者認證

1.6.7 將主機設定為NIS 用戶端

要實作透過NIS 進行的使用者管理，請在下一個步驟設定一個NIS 用戶端。本節僅描述用戶端的組態。如需使用YaST 來執行NIS 伺服器組態的說明，請參閱章25, 使用NIS。

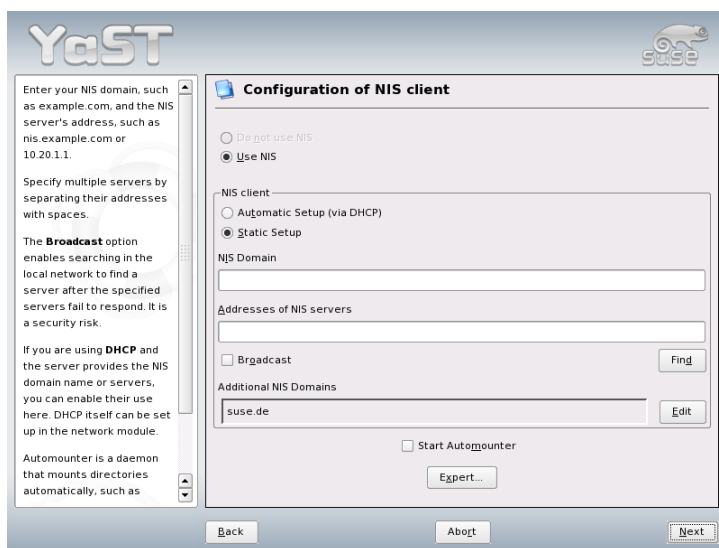


Figure 1.16: NIS 用戶端組態

在以下如圖形 1.16, “NIS 用戶端組態” 顯示的對話方塊中, 請先選取主機是否有靜態 IP 位址或是要透過 DHCP 取得一個。如果您選取 DHCP, 您無法指定 NIS 領域或 NIS 伺服器位址, 因為這些是由 DHCP 伺服器提供。如需 DHCP 的相關資訊, 請參閱章 27, DHCP。如果使用靜態 IP 位址, 請手動指定 NIS 領域和 NIS 伺服器。

要在網路中搜尋 NIS 伺服器的廣播, 請核取相關選項。您也可以設定多個 NIS 領域以及指定一個預設領域。對於每一個領域, 請選取“編輯”來指定多個伺服器位址, 或者針對每一個領域啟用廣播功能。

在進階設定中, 請使用“僅回覆本地主機”來防止其他網路主機查詢您的客戶端使用哪一個伺服器。如果您啟用“中斷的伺服器”, 未具有權限連接埠上的伺服器送出的回應也會被接受。如需詳細資訊, 請參閱 `yplibind` 的 man 頁面。

1.6.8 建立本地使用者帳戶

如果您決定不使用驗證伺服器來進行使用者驗證, 請建立本地使用者。使用者帳戶相關的任何資料(姓名、登入、密碼等)都將儲存在已安裝的系統上並進行管理。

Linux 是一種容許同時有多使用者在同一系統上進行作業的作業系統。每一個使用者需要一個使用者帳戶來登入系統。透過使用者帳戶，系統可以執行重要的安全性管理。例如，一般使用者無法變更或刪除系統正常運作所需的檔案。與此同時，其他使用者也無法修改、檢視或竄改特定使用者的個人資料。每位使用者可以設定自己的工作環境，而且每次登入時不會有任何變更。

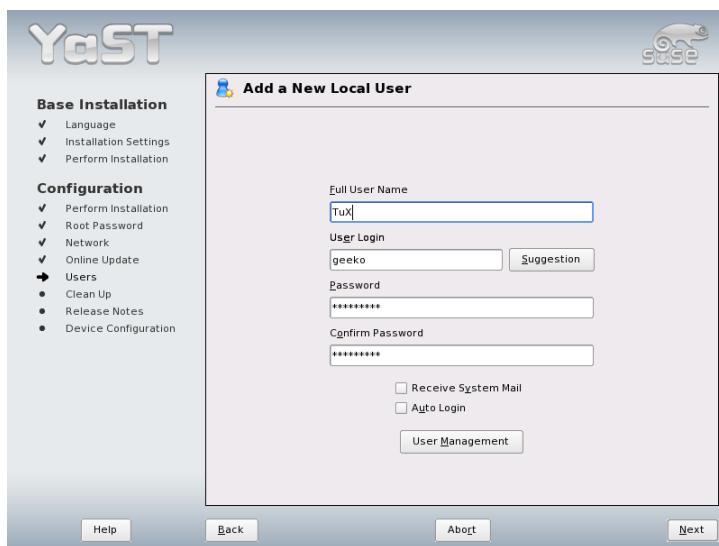


Figure 1.17: 輸入使用者名稱與密碼

使用者帳戶可以使用圖形 1.17, “輸入使用者名稱與密碼” 顯示的對話方塊來建立。輸入名字和姓氏後，請指定使用者名稱(登入)。可按一下‘建議’由系統自動產生使用者名稱。

最後，請輸入使用者的密碼。請再輸入一次進行確認(請確認沒有輸入錯誤)。使用者名稱可以通知系統使用者是誰，密碼則用來識別使用者身份。

┌

使用者名稱與密碼

請記住您的使用者名稱和密碼，因為每次登入系統時都會需要使用者名稱和密碼。

└

請設定5至8個字元的密碼來達到有效安全性。密碼的長度上限是128個字元。不過，如果未載入特殊的安全模組，只有開頭8個字元將被用於辨識密碼。密碼區分大小寫。不容許使用變化母音等特殊字元。其他特殊字元(7位元ASCII)以及數字0至9不在此限。

還有兩個額外選項可用於本地使用者：

‘透過電子郵件接收系統訊息’勾選這個方塊可以傳送系統服務建立的使用者訊息。這些訊息通常只傳送給root(根)，即系統管理員。這個選項對於使用頻繁的帳戶非常有用，因為最好在特殊的情況下才建議以root(根)身份登入。

‘自動登入’這個選項要在KDE當作預設桌面時才可使用。當系統啟動時，目前的使用者會自動登入系統。主要用於電腦只有一個使用者的情況。

└

自動登入

啓用自動登入時，系統開機時會直接進入桌面而無需驗證。如果您在系統上儲存敏感資料而且還有其他人存取本電腦的話，請不要啓用這個選項。

└

1.6.9 發行記事

完成使用者驗證設定後，YaST將顯示發行記事。建議您詳加閱讀，因為其中可能含有未列印在手冊中的最新資訊。如果您已經安裝更新套件，您會看到從SUSE伺服器擷取的最新版本發行記事。

1.7 硬體組態

在安裝程序的最後，YaST會開啓進行圖形卡以及系統連接之其他硬體元件，例如，印表機或音效卡的組態對話方塊。請按一下個別元件來啓動硬體組態。在大部份情況，YaST會自動偵測設備並進行組態。

您可以略過任何週邊設備並在以後進行組態。不過，您最好現在設定圖形卡。儘管YaST自動設定的顯示器設定值通常都可以被接受，但大部份的使用者對於解析度、色彩深度以及其他圖形功能卻有強烈的個人偏好。要變更這些設定，請選取‘圖形卡’。組態的說明位於節11.1, “使用SaX2來設定X11”。YaST完成組態資料寫入後，請使用最後一個對話方塊中的‘完成’來結束SUSE LINUX的安裝。

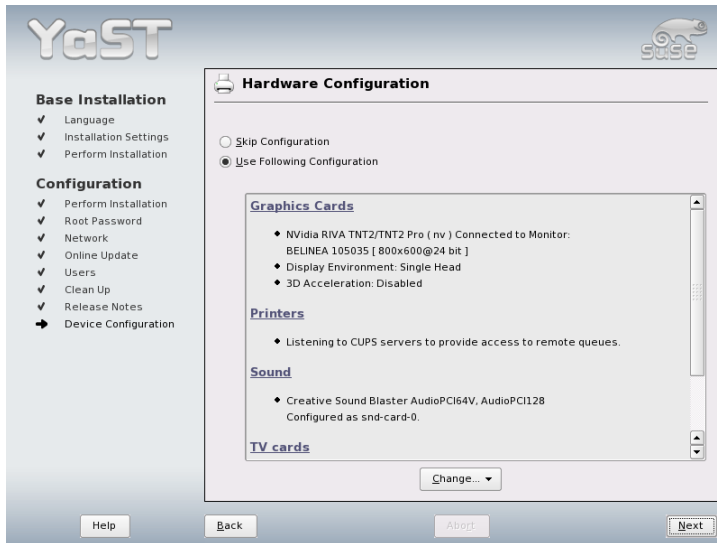


Figure 1.18: 設定系統元件

1.8 圖形登入

SUSE LINUX 現在已經完成安裝。如果您在本地使用者管理模組已經啓用自動登入，請直接啓動而不必登入。如果不是，您應該會在畫面上看到圖形登入，如圖形1.19, “KDM 的登入畫面” 所示。請輸入登入以及密碼來登入系統。

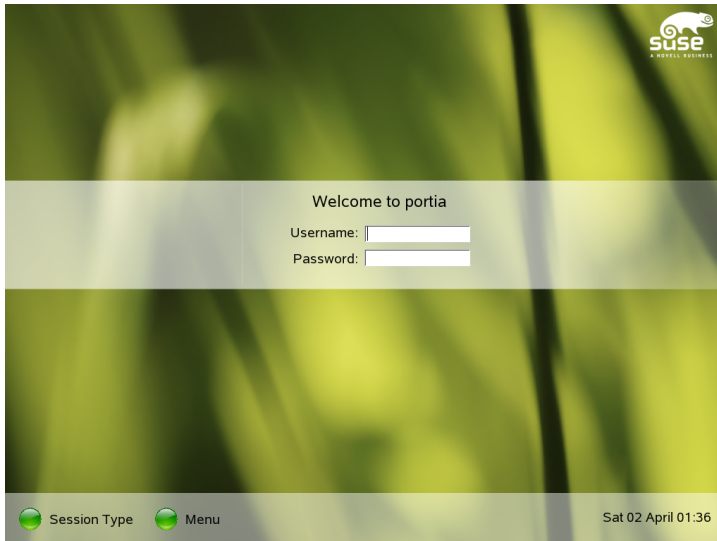


Figure 1.19: KDM 的 登 入 畫 面

使用YaST 進行系統組態

YaST 是可供安裝使用的設定工具，也是SUSE LINUX 的組態工具。本章內容涵蓋使用YaST 進行系統組態。這包括大部分硬體、圖形使用者介面、網際網路存取、安全性設定、使用者管理、軟體安裝、系統更新與系統資訊的設定。本章也提供在文字模式中使用YaST 的指示。

2.1	YaST 控制中心	32
2.2	軟體	33
2.3	硬體	44
2.4	網路設備	49
2.5	網路服務	50
2.6	保全性與使用者	52
2.7	系統	56
2.8	其他	64
2.9	文字模式(ncurses) 中執行YaST	65
2.10	從指令行進行線上更新	69

採用YaST的系統組態是利用許多不同的YaST模組進行。視硬體平台與安裝的軟體而定，共有三種方式可在安裝的系統中存取YaST。

在KDE或GNOME中，從SUSE功能表(‘系統’→‘YaST’)啟動「YaST控制中心」。此外，個別的YaST組態模組會整合至「KDE控制中心」。YaST啟動之前，會提示您輸入root密碼，因為YaST需具備系統管理員權限才能變更系統檔案。

若要從指令行啟動YaST，請輸入指令su(以變更為使用者root)與yast2。若要啟動文字版的YaST，請輸入yast而不是yast2。也可以使用指令yast從其中一個虛擬主控台來啟動程式。

若要變更YaST的語言，請在「YaST控制中心」中選擇‘系統’→‘選擇語言’。選擇語言之後，便可結束「YaST控制中心」，登出系統並再次登入。下次啟動YaST時，就會啓用新的語言設定。

對於無法支援自有顯示設備的硬體平台，以及要在其他主機進行遠端管理，就要從遠端執行YaST。首先，在要顯示YaST的主機上開啓主控台，然後輸入指令ssh -X root@<system-to-configure>以登入系統來設定root，並將X伺服器輸出重新導向至您的終端機。在成功登入SSH之後，輸入yast2以圖形模式啟動YaST。

若要在另一個系統以文字模式啟動YaST，請使用ssh root@<system-to-configure>來開啓連線。然後利用yast來啟動YaST。

2.1 YaST 控制中心

在圖形模式啟動YaST時，會開啓「YaST控制中心」，如圖形2.1, “YaST控制中心”所示。左邊的框架包含‘軟體’、‘硬體’、‘系統’、‘網路設備’、‘網路服務’、‘安全性與使用者’、‘系統’與‘其他’類別。如果按一下圖示，就會在右側列出內容。然後選擇想要的元件。例如，如果選擇‘硬體’，再按一下右邊的‘聲音’，就會開啓音效卡的組態對話方塊。個別項目的組態通常由數個步驟組成。請按‘下一步’以繼續進行下面的步驟。

大部分模組的左邊框架會顯示說明文字，解釋所需的項目。若要在沒有說明框架的模組中取得說明，請按(F1)或選擇功能表中的‘說明’。完成需要的設定之後，在最後一個組態對話方塊中按下‘完成’，就能完成程序。將組態儲存起來。

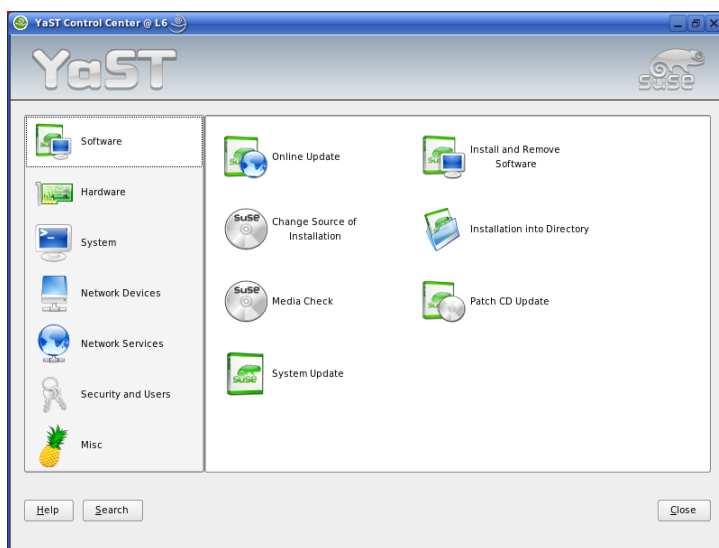


Figure 2.1: YaST 控制中心

2.2 軟體

2.2.1 安裝和移除軟體

此模組可讓您在電腦上進行安裝、解除安裝以及更新軟體。在Linux中，是以套件的形式取得軟體。一般情況下，套件會包含程式所需的一切：程式本身、組態檔案與文件。通常也可以取得包含程式來源檔案的套件。執行程式並不需要來源檔案，但您可能想要安裝來源以編譯程式的自訂版本。

部分套件依存於其他套件。這代表必須安裝另一個套件，才能正常運作該套件軟體。再者，也許是因為安裝常式需要特定的工具，必須安裝特定的其他套件後，才能夠安裝某些套件。因此，必須以正確的順序來安裝這類套件。某些套件會具有相同或相似的功能。如果這些套件會使用到相同的系統資源，就不應該同時進行安裝(套件衝突)。依存度與衝突會出現在兩個或多個套件之間，而且有時會相當複雜。為了讓互動順暢而要求特定的套件版本時，就會使情況更為複雜。

安裝、解除安裝與更新軟體時，必須考慮到所有因素。YaST為此提供非常有效率的工具：軟體安裝模組，通常稱為套件管理員。套件管理員啟動時，就會檢查

系統並顯示已安裝的套件。如果您選擇安裝其他套件，套件管理員就會自動檢查其依存度，並選擇其他所需的套件(依存度解決方案)。如果選擇相互衝突的套件，套件管理員就會加以指出，並提交可解決問題的建議(衝突解決方案)。如果已經標示要刪除其他已安裝套件所需的套件，套件管理員就會發出具有詳細資訊與替代方案的警告。

除了這些純技術面的觀點之外，套件管理員也能為SUSE LINUX 中涵蓋的套件提供結構良好的綜覽。按照主題來排列套件，並以適當的過濾器來限制這些群組的顯示。

套件管理員

若要使用套件管理員變更系統的軟體選擇，請在「YaST 控制中心」選擇「安裝或移除軟體」。套件管理員對話方塊視窗如圖形2.2, “YaST 套件管理員” 所示。該視窗包含不同的框架。按一下並移動分隔區域的直線就可以修改框架大小。框架內容與它們的用法描述如下。

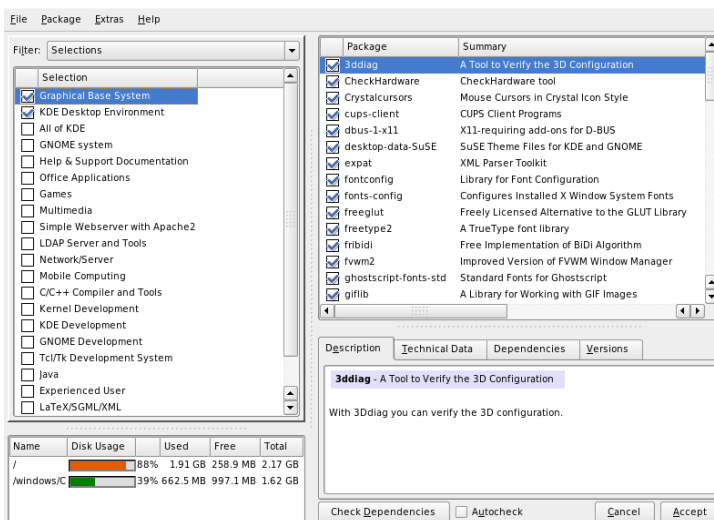


Figure 2.2: YaST 套件管理員

過濾器視窗

套件管理員可提供不同的過濾方法，依類別排列套件以及限制顯示的套件數目。過濾器視窗位在功能表列的左下方。可控制與顯示不同的過濾方法。上方的過濾器選擇方塊可決定顯示在過濾器視窗下方的套件。按一下過濾器選擇方塊，從可用過濾器清單中選擇過濾器。

選擇過濾器 啓動時，會啓用‘選擇’過濾器。此過濾器會根據應用程式用途，將程式套件分組，例如為多媒體或辦公室應用程式。不同的‘選擇’過濾器群組，都列在過濾器選擇方塊下方。安裝在系統上的套件，已事先選取。在行的開頭處按一下狀態方塊，以切換選擇的狀態旗標。直接以滑鼠右鍵在選項按一下並使用內容功能表，選擇狀態。右邊的個別套件視窗會顯示目前選擇中所包含的套件清單，讓使用者選取或取消選取個別套件。

套件群組過濾器 ‘套件群組’過濾器提供更多套件範圍的技術相關綜覽，適合熟悉SUSE LINUX 套件結構的使用者。此過濾器會在左邊的樹狀結構中，依主題排序程式套件，例如，應用程式、開發與硬體。展開的分支越多，選擇就越明確，而且在個別套件視窗右邊顯示的套件也越少。

此外，此過濾器也能夠在沒有分類的情況下，依字母順序顯示所有套件。要這樣做，請在最上層選擇‘zzz 全部’。由於SUSE LINUX 包含大量的套件，因此要顯示此份冗長的清單可能需要花費一些時間。

搜尋功能 ‘搜尋’功能是尋找特定套件最容易的方法。指定不同的搜尋標準，就可以有效限制過濾器，使個別套件視窗中經常只顯示一個套件。輸入搜尋字串並使用核取方塊，決定要在何處搜尋此字串(名稱、描述或套件依存度)。進階使用者可使用萬用字元與一般表示式來定義特殊搜尋型式，並在‘提供’與‘需要’欄位中搜尋套件依存度。例如，此功能可以用來判斷出包含特定程式庫的套件。

快速搜尋

除了‘搜尋’過濾器之外，套件管理員的所有清單都有快速搜尋的功能。只要輸入字母，就可以將游標移動到清單中以該字母為開頭名稱的第一個套件。游標必須位於清單中(按一下該清單即可)。

語言 某些SUSE LINUX 的套件中有可用的特定語言套件，例如程式使用者界面的翻譯文字、文件及字型等等。此過濾器會在左邊視窗中顯示所有SUSE LINUX 支援的語言清單。如果您選擇其中一種語言，則右邊視窗會顯示該

種語言適用的所有套件。其中套用至您目前軟體選項的所有套件會自動加上標籤以進行安裝。

Note

因為特定語言套件可能需要其他套件，因此有時候套件管理員會加選其他套件來安裝。

Note

安裝摘要 選擇要安裝、更新或刪除的套件之後，可使用過濾器選擇來檢視安裝摘要。摘要會顯示當您按一下‘接受’時，該如何處理套件。使用左邊的核取方塊來過濾套件，在個別的套件視窗中檢視。例如，若要檢查已經安裝哪些套件，請啟動套件管理員，只選取‘保留’並取消選取其它核取方塊。

可以像平常一樣，變更個別套件視窗中的套件狀態。不過，對應的套件可能會不再符合搜尋標準。若要移除清單中的這類套件，請使用‘更新清單’來更新清單。

個別套件視窗

如上所述，個別套件的清單會顯示在個別套件視窗的右側。此清單的內容，視目前選取的過濾器而定。例如，如果選取‘選擇’過濾器，個別套件視窗就會顯示目前選擇的所有套件。

在套件管理員中，每個套件的狀態可決定要如何處理該套件，例如「安裝」或「刪除」。在狀態方塊中，此狀態會在每行的開頭以符號顯示。在項目上按一下滑鼠右鍵，從功能表按一下或選擇想要的狀態，就可以切換狀態。根據目前的情況，可能無法選擇部分狀態旗標。例如，無法將尚未安裝的套件設為「刪除」。利用‘說明’→‘符號’檢視可用的狀態旗標。

套件管理員提供下列套件狀態旗標：

不安裝 此套件尚未安裝，而且將不會進行安裝。

安裝 此套件尚未安裝，但將會進行安裝。

保留 此套件已經安裝，而且將不會進行變更。

更新 此套件已經安裝，而且安裝媒體上的版本將會取代目前的版本。

刪除 此套件已經安裝，但是將會刪除。

禁用—永遠不要安裝 此套件尚未安裝，而且將永遠不會進行安裝。任何安裝媒體會將它視為不存在。如果為了解決依存度而會自動選擇該套件，將該套件設為「禁忌」就可以避免這種情形發生。不過，如此將會造成不一致，而必須手動解決問題(依存度檢查)。因此，「禁忌」主要適用於專業使用者。

保護 此套件已經安裝，而且不應有所修改。此狀態會自動指定給協力廠商套件(沒有SUSE簽名的套件)，避免套件被安裝媒體上較新的版本覆寫。這可能會造成套件衝突，而必須手動解決問題。

自動安裝 因為另一個套件需要此套件，所以已自動選取以進行安裝(套件依存度的解決方案)。若要取消選取這類套件，可能需要使用「禁忌」狀態。

自動更新 已經安裝此套件。不過，由於另一個套件需要此套件具備較新的版本，所以將會自動更新已安裝的版本。

自動刪除 此套件已經安裝，但是發生套件衝突的情形，因而必須刪除此套件。例如，如果已經使用其他套件取代目前的套件，就會造成這種情況。

自動安裝(選擇之後) 因為是預先定義的選擇，例如「多媒體」或「開發」，所以會自動選取要安裝此套件。

自動更新(選擇之後) 此套件已經安裝，但安裝媒體上有較新的版本。此套件是預先定義的選擇，例如「多媒體」或「開發」，已選取要進行更新，而且將會自動更新。

自動刪除(選擇之後) 此套件已經安裝，但預先定義的選擇(例如「多媒體」或「開發」)要求刪除此套件。此種情況不會經常發生。

此外，您可以決定是否要安裝套件的來源。此資訊可以補足目前的套件狀態，而且無法使用滑鼠來切換，或是直接從內容功能表來選擇。相反地，套件行結尾處的核取方塊才能讓使用者選擇來源套件。此選項也可以在‘套件’下方存取。

安裝來源 也要安裝原始程式碼。

不安裝來源 將不會安裝原始程式碼。

個別套件視窗中，不同套件所使用的字型顏色能提供其他資訊。安裝媒體上若有較新的版本可供已安裝的套件使用，就會顯示為藍色。已安裝的套件版本號碼若比安裝媒體上的版本更高，就會顯示為紅色。不過，由於套件的版本編號並非一直都是線性增加，因此資訊可能不夠完美，但也應該足以指出有問題的套件。如有需要，請檢查資訊視窗中的版本號碼。

資訊視窗

框架右下方的索引標籤，可提供有關所選取套件的許多資訊。所選取套件的描述會自動啓用。按一下其他索引標籤，可檢視技術資料(套件大小、群組等)、該套件依存的其他套件清單，或版本資訊。

資源視窗

選擇軟體時，左下方的資源視窗會顯示所有已裝載檔案系統的預期用量。每增加一個選擇，就會使色條的圖形增長。只要圖形仍是綠色，即代表有足夠的空間。隨著磁碟空間逐漸用盡，圖形的顏色會慢慢變成紅色。如果您選擇要安裝的套件過多，就會顯示警示。

功能表列

上述的大多數功能都可在視窗左上方的功能表列取得，其中包含四個功能表：

檔案 選擇‘檔案’→‘輸出’可將所有安裝套件的清單儲存在文字檔中。如果想要在日後或另一個系統上複製特定的安裝範圍，建議您這樣做。以此種方式產生的檔案可以使用‘輸入’來輸入，並產生與儲存清單相同的套件選擇。前文所述的兩種情況，都會定義檔案位置或接受建議。

若要結束套件管理員，但不儲存套件選擇的變更，請按一下‘結束—丟棄變更’。若要儲存變更，請按一下‘結束—儲存變更’。在此狀況下，會套用所有變更並終止程式。

套件 ‘套件’功能表中的項目，永遠會參照個別套件視窗目前選擇的套件。雖然會顯示出所有狀態旗標，但您只能選取目前套件可以使用的旗標。請使用核取方塊來決定是否要安裝套件來源。‘清單中所有套件’會開啓子功能表，列出所有套件狀態旗標。不過，這些不僅會影響到目前的套件，還會影響此清單中的所有套件。

額外 ‘額外’功能表可提供處理套件依存度與衝突的選項。如果手動選取要安裝的套件，請按一下‘顯示自動套件變更’來檢視套件管理員為了解決依存度問題，自動選取的套件清單。如果仍有未解決的套件衝突，就會顯示警示以及建議的解決方案。

如果將套件衝突設為‘忽略’，此項資訊就會永久儲存在系統中。否則，每次啓動套件管理員時都需要將相同的套件設為‘忽略’。若不要忽略依存度，請按一下‘重設忽略的依存度衝突’。

說明 ‘說明’→‘綜覽’可提供套件管理員功能的簡短解釋。不同套件旗標的詳細描述可在‘符號’下方取得。如果不要使用滑鼠來操作程式，請按一下‘按鍵’來檢視捷徑清單。

依存度檢查

‘檢查依存度’與‘自動檢查’位於資訊視窗下方。如果按一下‘檢查依存度’，套件管理員就會檢查目前的套件選擇是否會產生無法解決的套件依存度或衝突。如果有無法解決的依存度存在，就會自動選取所需的其他套件。若為套件衝突，套件管理員就會開啓對話方塊，顯示該衝突並提供解決問題的不同選項。

如果啓用‘自動檢查’，只要套件狀態有所變更就會觸發自動檢查作業。因為套件選擇的一致性永遠會受到監視，所以這個功能很實用。不過，此程序耗費資源，而且會使套件管理員的速度減緩。基於此原因，預設並不會啓用自動檢查功能。無論是哪種情況，當您利用‘接受’來確認選擇時，都會執行一致性檢查。

在下列範例中，可能無法同時安裝sendmail與postfix。圖形2.3, “套件管理員的衝突管理”顯示衝突訊息，提示您做決定。postfix已經安裝。因此，您應該避免安裝sendmail、移除postfix或承擔忽略該衝突的風險。

┌

處理套件衝突

處理套件衝突時，建議您遵循YaST的建議，否則存在的衝突可能會危及系統的穩定性與功能。

└

2.2.2 變更安裝來源

YaST可管理許多不同的安裝來源。這讓它們的選項可用於安裝或更新用途。當此模組啓動時，就會顯示所有先前已註冊的來源清單。從CD進行一般安裝之後，只會列出該安裝CD。按一下‘新增’，可以讓清單包含其他來源。您不但可以新增網路來源(例如，NFS與FTP伺服器)，還可以新增如CD與DVD的抽取式媒體。甚至可以選擇本地硬碟上的目錄，來作為安裝媒體。請參閱詳細的YaST說明文字。

所有已註冊的來源，在清單的第一欄都有啓動狀態。按一下‘啓動或關閉’以啓動或關閉個別的安裝來源。在安裝軟體套件或更新的期間，YaST會從啓動的安裝來源清單中選擇適用的項目。以‘關閉’結束模組時，就會儲存目前的設定，並套用到‘安裝及移除軟體’與‘系統更新’組態模組。

2.2.3 YaST 線上更新

「YaST 線上更新」(YOU)可讓使用者安裝重要的更新與改良。使用者可在SUSE FTP 伺服器及鏡像伺服器下載這些修補程式。

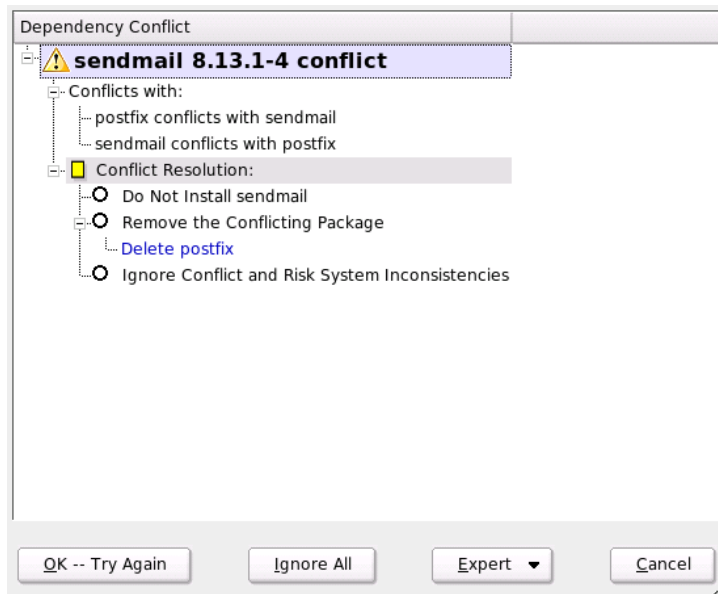


Figure 2.3: 套件管理員的衝突管理

在‘安裝來源’下方選擇其中一個伺服器。當您選擇伺服器時，伺服器的URL就會複製到輸入欄位，也可在此處編輯該URL。您也能以file:/my/path或/my/path的格式來指定本地的URL。使用‘新伺服器’可增加其他伺服器，擴充現有清單。按一下‘編輯伺服器’可修改目前選定伺服器的設定。

模組啟動時，會啟用‘手動選擇修補程式’，讓使用者選擇想要取得的修補程式。若要套用所有可用的更新套件，請關閉此選項。不過，根據連線的頻寬以及所要傳輸的資料量，如此將會造成下載時間過久。

如果啟用‘再次下載所有修補程式’，就會從伺服器下載所有可用的修補程式、可安裝套件與描述。如果未啟用此選項(預設)，就只會取回系統尚未安裝的修補程式。

此外，還可以自動更新系統。按一下‘設定全自動更新’以設定程序，定期自動尋找更新並予以套用。這是個完全自動化的程序。系統必須要在排定的時間，連線到更新伺服器。

若要執行更新，請按一下‘下一步’。若為手動更新，如此會載入所有可用的修補程式清單，並啟動套件管理員，請參閱節2.2.1, “安裝和移除軟體”。在套件管理

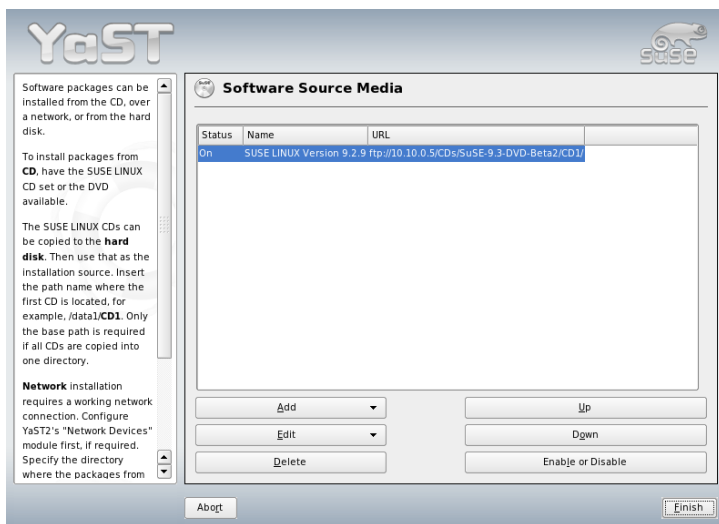


Figure 2.4: 變更安裝來源

員中會啓用YOU 修補程式的過濾器，讓使用者選擇要安裝的更新。啓動時，會事先選取可用的保安全性修補程式與建議使用的修補程式，前提是系統已經安裝相關的套件。使用者應該要接受此提案。

完成您的選擇之後，在套件管理員中按一下‘接受’。隨後就會從伺服器下載所有選取的更新，並在您的電腦上安裝。視連線速度與硬體效能而定，這可能會需要一些時間。如有錯誤則會顯示在視窗中。若有需要，請略過有問題的套件。安裝之前，部分套件會開啓顯示詳細資訊的視窗。

下載並安裝更新後，您可以在記錄視窗追蹤所有動作。成功安裝所有修補程式之後，按‘完成’可結束YOU。安裝之後如果不需要更新檔案，請選擇‘更新後移除來源套件’，在更新之後就會將來源套件刪除。最後，會視需要執行SuSEconfig來調整系統組態。

2.2.4 修補程式光碟更新

此選項會從光碟安裝修補程式，而不是從FTP 伺服器。使用光碟的優點是能更迅速進行更新。一旦插入修補程式光碟之後，就會掃描光碟上的所有修補程式，並顯示在對話方塊中。然後使用者可以從修補程式清單選取想要的套件進行安裝。

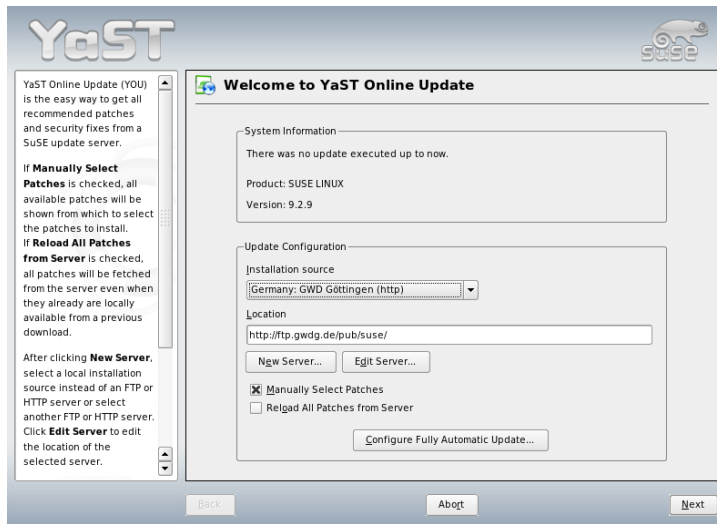


Figure 2.5: YaST 線上更新

如果修補程式光碟不存在，該模組就會發出錯誤訊息。請插入修補程式光碟，然後重新啓動該模組。

2.2.5 系統更新

此模組可以在系統上安裝更新的版本。操作期間只能更新應用程式軟體，不能更新SUSE LINUX基本系統。若要更新基本系統，請從安裝媒體將電腦開機，例如光碟。在YaST中選擇安裝模式時，請選擇‘更新現有系統’，而不是‘全新安裝’。

更新系統的程序與全新安裝類似。一開始，YaST會檢查系統，決定適當的更新策略，然後在建議對話方塊中顯示結果。以滑鼠按一下個別項目即可變更詳細資訊。例如，‘語言’與‘鍵盤配置’等部分項目，涵蓋在解釋安裝程序的章節中(請參閱節1.3, “語言選擇”)。本章節只涵蓋更新特定的設定。

選取以進行更新

如果系統上已經安裝數個SUSE LINUX版本，此項目可讓使用者從清單選擇分割區以進行更新。

更新選項

設定系統的更新方法。有兩個選項可用。

以安裝新軟體的方式更新 若要將整個系統更新至最新的軟體版本，請選擇其中一個預先定義的選項。這些選項與安裝期間所提供的選項相同。它們會確認先前不存在的套件也能進行安裝。

只更新已安裝套件 此選項只會更新系統上已存在的套件。將不會安裝任何新功能。

除此之外，您可以使用‘刪除過時套件’，移除新版本中不存在的套件。根據預設，會事先選取此選項，以避免過時的套件佔據硬碟空間。

套件

按一下‘套件’以啓動套件管理員，以及選取或取消選取要進行更新的個別套件。任何套件衝突應該利用一致性檢查來解決。套件管理員的使用會在節2.2.1, “安裝和移除軟體”中詳細說明。

備份

更新期間，部分套件的組態檔案可能會以新版本取代。因為您可能已修改過目前系統中的部分檔案，所以套件管理員通常會針對被取代檔案進行備份。請利用此對話方塊來決定備份的範圍。

備份範圍

此備份並不包括軟體。僅包含組態檔案。

更新的重要資訊

系統更新是非常複雜的程序。對於每個程式套件而言，YaST 必須先檢查電腦上安裝的版本，然後決定需要進行的操作，才能以新的版本正確取代舊版套件。YaST 也會嘗試套用已安裝套件的任何個人設定。因為舊的組態無法順利處理新的程式版本，或是因為不同的組態之間意外出現不一致之處，所以可能有部分組態會引發一些問題。

現有的版本越舊，需要更新的套件組態與標準間的分歧越多，進行更新將會產生更多問題。有時候會無法正確套用舊組態。在此狀況下，則必須進行全新組態。開始進行更新之前，應該要另外儲存現有的組態。

2.2.6 媒體檢查

如果您使用SUSE LINUX 安裝媒體時遇到任何問題，您可以用此模組來檢查CD或DVD。在少數的情況下，某些設備可能無法正確讀取某些媒體。最有可能是「自製」的媒體。要檢查SUSE LINUX CD 或DVD 是否有錯誤，只要將該媒體放入設備中並執行此模組。按一下‘啓動’然後YaST 會檢查媒體的MD5 檢查總數。這可能會花費幾分鐘。如果偵測到錯誤，您就不應該使用此媒體來進行安裝。

2.3 硬體

必須先依照廠商指定的方式來安裝或連接新的硬體。開啓外部設備，例如印表機或數據機，然後啓動對應的YaST 模組。YaST 能自動偵測到大部分的設備，並顯示技術資料。如果自動偵測失敗，YaST 就會提供設備清單(型號、廠商等)，讓使用者由此選擇適當的設備。如需詳細資訊，請參閱硬體隨附的文件。

指定型號

如果設備清單中沒有您的型號，請嘗試選擇名稱類似的型號。不過，由於相似的名稱不一定代表具備相容性，所以在某些情況下必須與型號完全相符。

2.3.1 CD-ROM 與DVD 光碟機

在安裝範圍內，所有偵測到的CD-ROM 光碟機會利用/etc/fstab 檔案中的項目，整合到已安裝的系統中。/media 中會建立對應的子目錄。請使用此YaST 模組，將其他光碟機整合到系統。

啓動模組之後，就會以清單顯示所有偵測到的光碟機。使用核取方塊在行的開頭處標示您的新光碟機，然後利用‘完成’來完成整合作業。接著就會將新光碟機整合到系統。

2.3.2 印表機

在Linux 中列印的詳細資訊，請參閱章12, 印表機操作，它也含涵蓋一般列印問題。YaST 會自動設定印表機或提供組態對話方塊來協助手動設定印表機。然後

您可以從指令列來列印，或設定應用程式來使用列印系統。如需YaST 中關於設定印表機的詳細說明，請參閱節12.5.1, “本地印表機”。

2.3.3 硬碟控制器

通常YaST 會在安裝期間設定系統的硬碟控制器。如果要新增控制器，請使用此YaST 模組整合到系統。您也可以修改現有的組態，但一般而言不需要這樣做。

對話方塊會以清單顯示所偵測到的硬碟控制器，可讓使用者利用特定的參數指定適合的核心模組。在系統中設定永久性儲存之前，請使用‘測試模組載入’來檢查目前的設定是否能夠運作。

↑

硬碟控制器的組態

這是一種專業工具。如果所作的設定不正確，您的系統可能會無法再度開機。如果設定有所變更，請使用測試選項。

↓

2.3.4 硬體資訊

YaST 可為硬體元件偵測其硬體組態。所偵測到的技術資料會以個別的畫面顯示。如果您需要有關硬體的資訊，而想要提交支援要求時，這就會特別實用。

2.3.5 IDE DMA 模式

利用此模組，為已安裝系統的IDE 硬碟以及IDE CD 與DVD 光碟機，啟動和關閉DMA 模式。此模組對SCSI 設備沒有任何作用。DMA 模式能夠大幅提昇系統的效能與資料傳輸速度。

在安裝期間，目前的SUSE LINUX 核心會自動啟動硬碟的DMA，但不會啟動CD 光碟機的DMA，因為預設啟動所有磁碟機的DMA，經常會造成CD 光碟機方面的問題。請使用DMA 模組來啟動磁碟機的DMA。如果磁碟機可支援DMA 模式而不會發生問題，啟動DMA 將可提昇磁碟機的資料傳輸速度。

DMA (直接記憶體存取) 表示您的資料可以跳過處理器控制，直接傳輸到RAM。

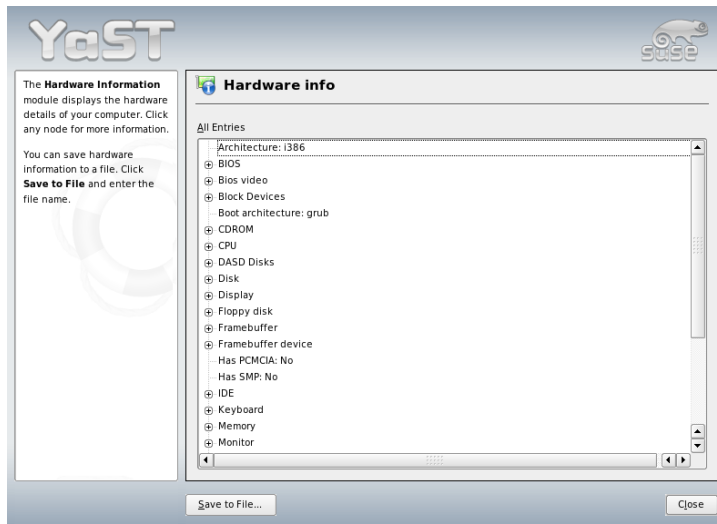


Figure 2.6: 顯示硬體資訊

2.3.6 掃描器

如果已連接掃描器並開啓電源，此YaST 模組啓動時應可自動偵測到。在此狀況下，就會出現安裝掃描器的對話方塊。如果沒有偵測到掃描器，就會出現手動組態對話方塊。如果已經安裝一或數台掃描器，就會顯示現有掃描器的清單以供修改或刪除。按下‘新增’以設定新的設備。

接下來，就會以預設設定來執行安裝。如果順利完成安裝，就會出現對應的訊息。現在，請放入文件並按一下‘測試’，來測試您的掃描器。

未偵測到掃描器

只會偵測到可支援的掃描器。而無法偵測到連線到另一個網路主機的掃描器。手動組態將掃描器區分成三種類型：USB 掃描器、SCSI 掃描器與網路掃描器。

USB 掃描器 請指定廠商與型號。接著，YaST 會嘗試載入USB 模組。如果是非常新型的掃描器，該模組可能無法自動將模組載入。在此狀況下，會自動進行使用者手動載入USB 模組的對話方塊。如需詳細資訊，請參閱YaST 說明文字。

SCSI 掃描器 請指定該設備，例如 `/dev/sg0`。系統正在執行時，不應該連接SCSI 掃描器或解除連接。請先關閉系統。

網路掃描器 輸入IP 位址或主機名稱。若要設定網路掃描器，請參閱「支援資料庫」文章在Linux 中進行掃描 (<http://portal.suse.com/sdb/en/index.html>，關鍵字為掃描器)。

如果未偵測到您的掃描器，可能是該設備不受支援。不過，即使是受支援的掃描器，有時也會偵測不到。如果是該狀況的話，請利用手動掃描器選擇來繼續進行。如果可以在廠商與型號清單中找到您的掃描器，請加以選擇。如果找不到，請選擇‘取消’。有關掃描器與Linux 搭配使用的資訊，請參閱<http://cdb.suse.de/> 與<http://www.mostang.com/sane>。

┌

手動指定掃描器

在您完全有把握時，才以手動的方式來指定掃描器。不正確的選擇會使您的硬體有所損壞。

└

疑難排解

由於下列其中一種原因，可能會偵測不到您的掃描器：

- 未支援該掃描器。請檢查<http://cdb.suse.de/> 的Linux 相容設備清單。
- 未正確安裝SCSI 控制器。
- 您的SCSI 連接埠有終止問題。
- SCSI 纜線太長。
- 該掃描器含有Linux 不支援的SCSI 燈號控制器。
- 該掃描器有損壞。

┌

系統正在執行時，不應該連接SCSI 掃描器或解除連接。請先關閉系統。

└

如需掃描的相關資訊，請參閱用戶指南 中有關kooka 的章節。

2.3.7 聲音

啓動聲音組態工具之後，YaST 會嘗試自動偵測您的音效卡。設定一或多片音效卡。若要使用多片音效卡，請選擇其中一片介面卡，開始進行設定。按下‘設定’以進入‘設定’對話方塊。‘編輯’會開啓對話方塊，以編輯先前設定好的音效卡。‘完成’會儲存目前的設定，並完成聲音組態。

如果YaST無法自動偵測您的音效卡，請按下‘聲音組態’中的‘新增音效卡’，開啓對話方塊以選取音效卡與模組。請參閱音效卡文件，取得所需的資訊。ALSA 支援的音效卡參照清單以及對應的聲音模組，可在 `/usr/share/doc/packages/alsa/cards.txt` 與 <http://www.alsa-project.org/~goemon/> 取得。完成選擇之後，請按‘下一步’以回到‘設定’。

設定

請在第一個設定畫面中選擇組態層級。利用‘快速自動設定’，就無須通過任何進一步的組態步驟，而且不會執行聲音測試。可自動完成設定音效卡。利用‘一般設定’，您就可以調整輸出音量以及播放測試聲音。‘進階設定’可讓您手動自訂音效卡選項。

您也可以在此對話方塊中，找到搖桿組態的捷徑。按一下對應的核取方塊。在下列對話方塊中選擇搖桿類型，然後按‘下一步’。

音效卡音量

請在此測試畫面中，測試您的聲音組態。使用‘+’與‘-’來調整音量。爲了避免損壞您的喇叭或聽覺，請從百分之十的音量開始。當您按下‘測試’時，應該要能夠聽見測試聲音。如果聽不見，請增加音量。按下‘繼續’可完成聲音組態。接著就會儲存音量設定。

聲音組態

使用‘刪除’可移除音效卡。停用 `/etc/modprobe.d/sound` 檔案中已設定現有的音效卡項目。按一下‘選項’可開啓對話方塊，於此手動自訂聲音模組選項。其他音效卡可在‘新增音效卡’下方設定。如果YaST偵測到另一片音效卡，就會繼續進行到‘設定音效卡’。如果YaST未偵測到音效卡，就會自動導向‘手動音效卡選擇’。

如果您使用Creative Soundblaster Live 或AWE 音效卡，請利用‘安裝音色’從原始的Soundblaster 驅動程式光碟，將SF2 音色複製到您的硬碟。該音色會儲存在 `/usr/share/sfbank/creative/` 目錄中。

若要播放MIDI 檔案，請啓動‘啓動編曲程式’。如此一來，編曲程式支援的模組就會與聲音模組一同載入。

當您按一下‘完成’時，就會儲存所有已安裝的音效卡音量與組態。混音器設定會儲存到/etc/asound.conf 檔案，而ALSA 組態資料則會附加在/etc/modprobe.conf 檔案的結尾處。

2.3.8 電視與廣播卡

啓動與啓始化此YaST 模組之後，就會出現‘電視與廣播卡’對話方塊。如果自動偵測到您的介面卡，該介面卡就會出現在清單的上方。在此狀況下，請以滑鼠反白標示該行，然後選擇‘設定’。如果未偵測到您的介面卡，請選擇‘其他(無法辨識)’。按下‘設定’以繼續進行手動選擇，從廠商與型號清單中選擇您的介面卡。

如果已經設定電視或廣播卡，可利用‘變更’來修改現有的組態。在此狀況下，對話方塊會將所有已設定的介面卡顯示出來。選擇介面卡，然後利用‘編輯’開始進行手動組態。

在自動偵測硬體期間，YaST 會嘗試將正確的選台器指定給您的介面卡。如果您不太確定，只要保留‘預設(已辨識)’設定，然後檢查是否能夠運作即可。如果您無法設定所有頻道，這可能是因為在自動偵測選台器類型時發生錯誤。在此狀況下，請按一下‘選擇選台器’然後將清單中正確的選台器類型反白標示。

如果您熟悉技術的詳細資料，就可以使用專業對話方塊來指定電視或廣播卡的設定。在此對話方塊中選擇核心模組及其參數。也要檢查電視卡驅動程式的所有參數。要這樣做，請選擇對應的參數，並在參數行中輸入新的值即可。按一下‘套用’確認要使用新的值，或按一下‘重設’還原預設值。

‘電視與廣播卡、音效’對話方塊，可讓您利用已安裝的音效卡連接到電視或廣播卡。您必須利用音效卡的外部音效輸入，使用纜線來連接電視或廣播卡的輸出。這只有在已設定好音效卡，而且也啓用外部輸入的情況下才行得通。如果尚未設定您的音效卡，請選擇‘設定音效卡’可前往對應的對話方塊，請參閱節2.3.7, “聲音”。

如果您的電視或廣播卡有喇叭插孔，您也可以直接連接喇叭，而不必設定音效卡。市面上也有沒有聲音功能；也不需要音效組態的電視卡，例如CCD 監視攝影機。

2.4 網路設備

系統的所有網路設備必須在服務使用它們之前起始。這些設備的偵測和設定作業會在模組群組‘網路設備’中完成。在YaST 中設定支援的網路介面卡類型說明，包括連線到網路的相關背景資訊在內，請參閱節22.4, “使用YaST 手動設定網路連線”。關於無線通訊的網路設備組態，在章17, 無線通訊中有詳細的說明。

2.5 網路服務

此群組包含一些工具，用來設定網路中所有類型的服務。這些工具包含名稱解析、使用者驗證以及檔案服務。

2.5.1 郵件轉送代辦程式

如果您是利用sendmail、postfix 或提供者的SMTP 伺服器來傳送電子郵件，此模組可設定您的郵件設定。您可以透過fetchmail 程式取得郵件，也可以輸入提供者之POP3 伺服器或IMAP 伺服器的詳細資訊。或是，使用您選擇的郵件程式，例如KMail 或Evolution，如往常一樣設定POP 與SMTP 存取資料(利用POP3 接收郵件，以及利用SMTP 傳送郵件)。在此狀況下，您不需要此模組。

若要利用YaST 設定您的郵件，請在電子郵件設定模組的第一個對話方塊中，指定用來連線到網際網路的連線類型。選擇下列其中一個選項：

‘永久’ 如果您已有連到網際網路的專屬線路，請選擇此選項。您的電腦永久與網際網路連線，因此不需要撥號連線。如果您的系統位於使用中央電子郵件伺服器的本地網路中，請選擇此選項以確保可永久存取您的電子郵件。

‘撥號連線’ 有電腦在家中、電腦不在網路中，而且偶爾才會連線到網際網路的使用者才適合使用此項目。

無連線 如果您無法存取網際網路，而且電腦不在網路中，您就無法傳送或接收電子郵件。

因此，啓用對應的核取方塊，就可以利用AMaViS 為內送與外送的電子郵件，啓動病毒掃描功能。該套件會在您啓動郵件過濾功能之後，立即自動安裝。請在下列對話方塊中，指定外送郵件伺服器(通常是提供者的SMTP 伺服器)，以及內送郵件的參數。如果使用撥號連線，請根據不同的使用者，針對郵件接收個別指定不同的POP 或IMAP 伺服器。利用此對話方塊，您也可以指定別名、使用偽裝，或設定虛擬領域。按一下‘完成’ 以結束郵件組態。

2.5.2 其他可用的服務

YaST 中有很多其他的網路模組。

DHCP 伺服器 YaST 只需要幾個步驟，就能夠完成設定自訂的DHCP 伺服器。章27, DHCP 可提供該主題的基本知識，還會逐步說明YaST 中的組態程序。

DNS 伺服器 負責進行名稱解析的DNS 伺服器組態，建議您用在大型網路上。採用YaST 模組的組態，請參閱節24.1, “使用YaST 進行設定”。章24, 領域名稱系統提供有關DNS 的相關背景資訊。

DNS 和主機名稱 如果設定網路設備時，沒有進行這些設定，請使用此模組來設定主機名稱和DNS。此模組也可以用來變更主機名稱與領域名稱。如果提供者已經正確設定DSL、數據機或ISDN 存取，從提供者資料擷取出的項目，就會包含在名稱服務清單上。如果您處於本地網路中，可能會透過DHCP 收到您的主機名稱，若為此狀況，則不應修改該名稱。

HTTP 伺服器 若要執行您自己的網頁伺服器，請利用YaST 來設定Apache。如需詳細資訊，請參閱章30, Apache 網頁伺服器。

主機名稱 在開機時且在小型網路中，主機名稱解析也可以利用此模組來完成，而不必使用DNS。在此模組中的項目會反映/etc/hosts 檔案的資料。如需更多資訊，請參閱節22.5.1, “/etc/hosts”。

LDAP 用戶端 網路中的使用者驗證能以LDAP 來取代NIS。LDAP 的背景資訊以及利用YaST 的用戶端模組的詳細說明，請參閱章29, LDAP — 一種目錄服務。

NFS 用戶端與NFS 伺服器 NFS 可執行檔案伺服器，讓網路上的所有成員皆能存取。此檔案伺服器可讓使用者能夠使用特定應用程式、檔案與儲存空間。在‘NFS 伺服器’ 模組中，您可以將主機設定為NFS 伺服器，並決定要輸出的目錄，以供網路使用者用於一般用途。所有具有適當權限的使用者，都可以在他們自己的檔案樹中裝載這些目錄。如需YaST 模組的說明以及有關NFS 的背景資訊，請參閱章26, 使用NFS 共用檔案系統。

NIS 用戶端與NIS 伺服器 如果有一個以上的系統，本地使用者管理(使用/etc/passwd 與/etc/shadow) 不夠實際，而且需要花費許多心力維護。在此狀況中，應該要在中央伺服器上管理使用者資料，再從配送給用戶端。NIS 是可行的解決方案，另外還有LDAP 與Samba。有關NIS 與採用YaST 之組態的詳細資訊，請參閱章25, 使用NIS。

NTP 用戶端 NTP (網路時間協定) 這項協定可用來同步化網路上的硬體時鐘。有關NIS 的背景資訊與採用YaST 之組態的說明，請參閱章28, 使用xntp 進行時間同步化。

網路服務(inetd) 使用此工具可決定SUSE LINUX 開機時所要啓動的網路服務(例如finger、talk 與ftp)。這些服務可讓外部主機連線到您的電腦。每個服務都可以設定不同的參數。根據預設，並不會啓動管理個別服務(inetd 或xinetd) 的主要服務。

此模組啓動之後，請選擇要啓動inetd 或xinetd。選擇的精靈會以標準的服務選擇來啓動。或者，您可以利用‘新增’、‘刪除’與‘編輯’來編撰您自己的服務選擇。

└

設定網路服務(inetd)

在系統上撰寫與調整網路服務的程序複雜，必須要對Linux 服務的概念有廣泛的了解。

└

Proxy 您可以利用此模組來編輯全系統proxy 設定。您可以在章33, Proxy 伺服器：Squid 找到有關Proxy 的詳細資訊。

從遠端主機進行管理 若要從遠端主機透過VNC 連線來維護您的系統，請允許利用此YaST 來建立連線。請參照節3.2.2, “要安裝VNC 的用戶端”。

路由 如果您透過本地網路中的閘道來連線到網際網路時，就需要這個工具。對於DSL 而言，只需要有閘道資料，就可以設定網路卡。不過，DSL 的項目幾乎都是空的，沒有任何功能。

設定Samba 伺服器與用戶端 在Linux 與Windows 主機所組成的異質網路中，Samba 可控制兩方之間的通訊。有關Samba 的資訊以及用戶端與伺服器的組態，請參閱章32, Samba。

2.6 保全性與使用者

多重使用者是Linux 的基本功能。因此，多位使用者可以在相同的Linux 系統上獨立進行工作。登入名稱與個人密碼可用來識別每位使用者的使用者帳戶，以供登入系統使用。所有使用者都會擁有自己的主目錄，用以儲存個人檔案與組態。

2.6.1 使用者管理

選擇要編輯的使用者之後，YaST 會提供系統中所有本地使用者的綜覽。如果是大型網路的一部份，請按一下‘設定過濾器’，列出所有系統使用者(例如，root)或NIS 使用者。您也可以建立自訂的過濾器設定。除了在個別使用者群組間切換之外，也可以根據您的需要來合併群組。若要新增新的使用者，請填寫下列畫面中的必要欄位。隨後，新使用者就能利用登入名稱與密碼來登入主機。使

用‘詳細資訊’，可以針對使用者設定檔進行更詳細的設定。您可以手動設定使用者ID、主目錄以及預設的登入外圍程序。為新使用者指定特定群組。在‘密碼設定’中設定密碼的有效性。按一下‘編輯’可在有需要時，隨時變更這些設定。若要刪除使用者，請選擇清單上的使用者，然後按一下‘刪除’。

如需進階的網路管理，請使用‘進階選項’來定義建立新使用者的預設設定。選擇驗證方法(NIS、LDAP、Kerberos 或Samba) 以及密碼加密所要使用的演算法。這些設定與大型網路相關。

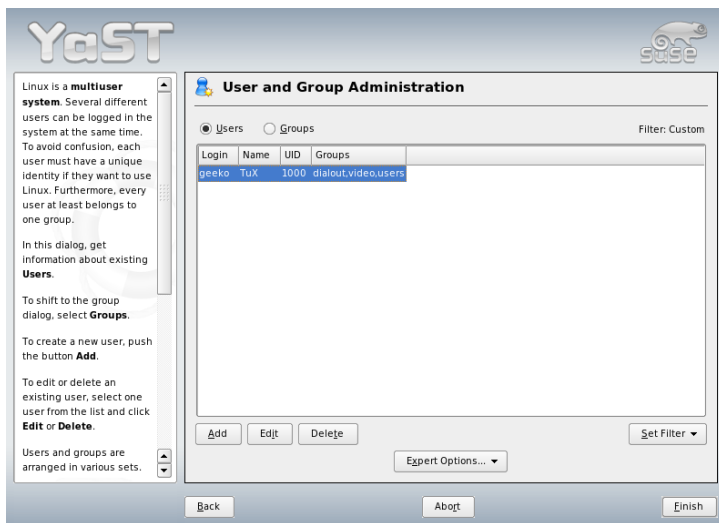


Figure 2.7: 使用者管理

2.6.2 群組管理

從「YaST 控制中心」啟動群組管理模組，或是在使用者管理中按一下‘群組’。這兩個對話方塊擁有相同的功能，都可讓您建立、編輯或刪除群組。

YaST 可提供所有群組的清單。若要刪除群組，請從清單中選取(選擇的行會以深藍色反白標示)，然後按一下‘刪除’。若要‘新增’以及‘編輯’，請在對應的YaST 畫面中輸入名稱、群組ID (gid) 與群組成員。若有需要，請針對此群組設定變更的密碼。過濾器設定與‘使用者管理’對話方塊相同。

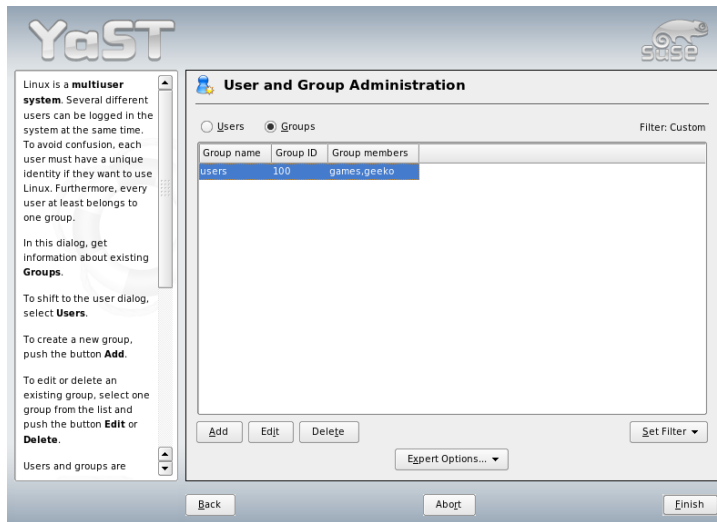


Figure 2.8: 群組管理

2.6.3 安全性設定

請在‘保全性與使用者’下的‘本地保全性組態’中，選擇下列四個選項的其中一個：層級1 供獨立電腦使用(預先設定)。層級2 供有網路的工作站使用(預先設定)。層級3 供有網路的伺服器使用(預先設定)。如需您自己的組態，請使用‘自訂設定’。

如果按一下前三個項目的其中一個，就會整合其中一個預先設定系統安全性選項的層級。要這樣做，請按一下‘完成’。在‘詳細資訊’之下能夠存取可供修改的個別設定。如果選擇‘自訂設定’，請按‘下一步’繼續進行不同的對話方塊。此處可找到預設的安裝值。

‘密碼設定’ 接受新的密碼之前，若要系統檢查密碼，請選擇‘檢查新密碼’與‘密碼的可信度測試’。為新建立的使用者設定密碼長度的上下限。定義密碼的有效期間，以及應該在到期前幾天內，於該使用者登入文字主控台時就發出警示。

‘開機設定’ 選擇想要的動作，指定解譯按鍵組合(Ctrl)-(Alt)-(Del)的方式。通常在文字主控台中輸入此組合，就會讓系統重新開機。除非您的電腦或伺服器

可供公用存取，而且您擔心會有人未經過授權就執行此動作，否則請不要修改此設定。如果選擇‘停止’，此按鍵組合就會使系統關機。使用‘忽略’，則會忽略此按鍵組合。

授與權限來指定‘KDM 的關機行為’，從KDE 顯示管理員(KDE 的圖形化登入) 關閉系統。可將權限授與‘只有root’(系統管理員)、『所有使用者’、『無人’或‘本地使用者’。如果選擇‘無人’，就只能透過文字主控台來關閉系統。

‘登入設定’ 一般情況下，登入失敗之後會先等待數秒，然後才能進行另一次登入。如此可讓密碼監聽程式(Sniffer) 不容易登入。可選擇性啓用‘記錄失敗登入次數’ 與‘記錄成功登入次數’。如果您懷疑有人試圖要找出您的密碼時，請在/var/log 中檢查系統記錄檔中的項目。利用‘允許遠端圖形登入’，其他使用者就可以透過網路取得圖形登入畫面的存取權。因為此存取方式有潛在的安全性風險，因此預設會關閉該功能。

‘新增使用者設定’ 每位使用者都擁有數值與字母混合的使用者ID。之間的關聯是透過/etc/passwd 檔案建立，而且應該是唯一的。使用此畫面中的資料，可在新增使用者時，針對要指定給使用者ID 的數值部分來定義數字範圍。使用者適用的下限為500；自動產生的號碼會從1000 開始。群組ID 設定請以相同的方式進行。

‘其他設定’ 對於‘檔案許可權設定’ 而言，共有三種選項：‘簡易’、『保全’與‘Paranoid’。第一種選項對於大部分使用者而言應已足夠。YaST 說明文字會提供三種安全性層級的相關資訊。‘Paranoid’ 設定相當嚴格，而且可作為系統管理員設定的作業基礎。請記得，如果選擇‘aranoid’，有些程式可能就無法運作或無法正確運作，因為使用者可能已經不具備存取特定檔案的權限。

在此對話方塊中，也可以定義哪些使用者可以啓動updatedb 程式。此程式每天都會自動執行，或是在開機後執行，您電腦上每個檔案的儲存位置都會包含在所產生的資料庫(locatedb) 內。如果選擇‘無人’，則使用者都只能夠在資料庫中，找到其他(未經授權) 使用者都能看到的路徑。如果選擇root，則會製作所有本地檔案的索引，因為root 使用者是超級使用者，可以存取所有目錄。最後，請確定已經關閉‘根路徑中目前的目錄’(預設)。

按下‘完成’ 來完成保安全性組態。

2.6.4 防火牆

使用此模組來設定SuSEfirewall2，可保護您的電腦不會受到來自網際網路的攻擊。有關SuSEfirewall2 的詳細資訊，請參閱節34.1, “偽裝與防火牆”。



Figure 2.9: 保全性設定

自動啓動防火牆

YaST 會根據每個已設定的網路介面，以合適的設定自動啓動防火牆。如果想要以自訂設定重新設定防火牆，或是關閉防火牆功能，您只需要啓動此模組。

2.7 系統

2.7.1 系統區域的備份複本

YaST 備份模組可讓您建立系統的備份。由此模組建立的備份並不包含整個系統。它只會儲存已變更套件的相關資訊，以及重要儲存區域的複本與組態檔案。定義要儲存在備份中的資料種類。根據預設，備份會包含上次安裝之後已變更套件的相關資訊。此外，它可能會包含不屬於套件的資料，例如 `/etc` 中的許

多組態檔案，或是/home 下的目錄。除此之外，備份還會包含硬碟上重要的儲存區域，該區域在嘗試還原系統時不可或缺，例如，分割區表或主要開機磁區(MBR)。

2.7.2 還原系統

如圖形2.10, “啓動還原模組的視窗” 所示的還原模組，可讓您從備份歸檔還原系統。請遵循YaST 中的指示。請按‘下一步’以繼續進行個別的對話方塊。首先，指定歸檔的位置(抽取式媒體、本地硬碟或網路檔案系統)。會顯示個別歸檔的說明與內容，讓您決定要從哪個歸檔來進行還原。

此外，會有對話方塊可將自從上次備份之後增加的套件解除安裝，還可以重新安裝自從上次備份之後刪除的套件。這兩個步驟可讓您將系統確實還原到上次備份時的狀態。

└

系統還原

因為此模組通常會安裝、取代或解除安裝許多套件與檔案，若您具有備份的經驗，才使用此模組。否則可能會遺失資料。

└

2.7.3 建立開機磁片與救援磁片

使用此YaST 模組來建立開機磁片與救援磁片。系統的開機組態損壞時，這些磁片非常有用。根分割區的檔案系統損壞時，格外需要救援磁片。

下列是可用的選項：

‘標準開機磁片’ 使用此選項以建立標準開機磁片，用來啓動已安裝的系統。依據結構的不同，實際的開機磁片數量可能會有所不同，但您應該建立對話方塊中所有的開機磁片，因為開機時需要這些開機磁片。啓動救援系統時也需要這些開機磁片。

‘救援磁片’ 此磁片包含的特殊環境，可讓您在已安裝的系統上執行維護工作，例如檢查與修復檔案系統，以及更新開機載入程式。若要啓動救援系統，請利用標準開機磁片來開機，然後選擇‘手動安裝’→‘啓動安裝或系統’→‘救援系統’。然後會提示您插入救援磁片。

‘自訂磁片’ 將現有的磁片影像從硬體寫入磁片。

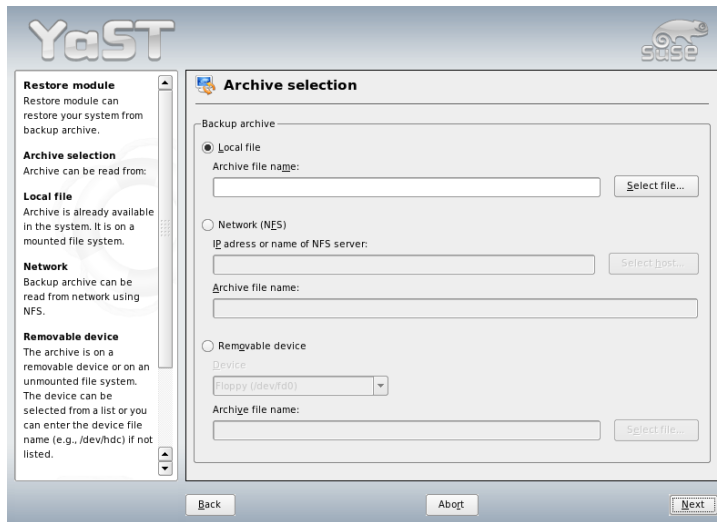


Figure 2.10: 啓動還原模組的視窗

‘下載磁片影像’ 使用此項目，請輸入URL 與驗證資料，從網際網路下載磁片影像。

若要建立其中一種磁片，請選擇對應的選項，再按‘下一步’。看到提示時插入磁片。如果再按‘下一步’，就會建立磁片。

2.7.4 LVM

Logical Volume Manager (LVM) 這項工具能利用邏輯磁碟機自訂硬碟的磁碟分割。如需LVM 的詳細資訊，請參閱節3.7, “LVM 組態”。

2.7.5 磁碟分割

您可使用進階對話方塊，如圖形2.12, “YaST 專家分割程式” 所示，手動修正一或多個硬碟分割。分割區可以新增、刪除或編輯。從此YaST 模組還可以存取軟體式RAID 與LVM 組態。

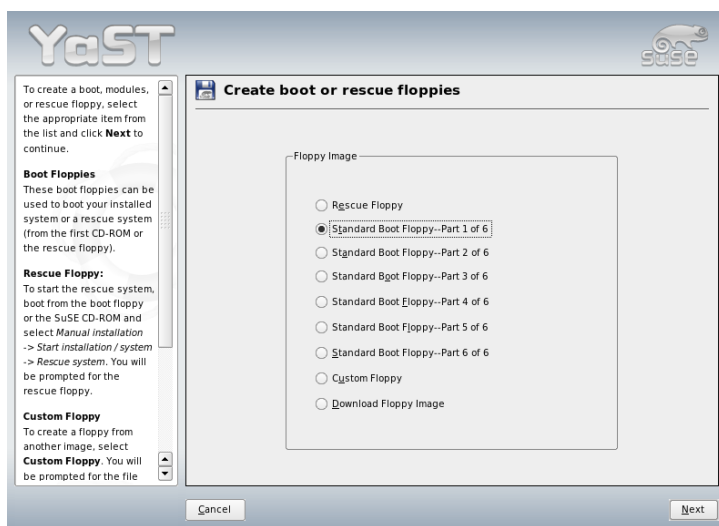


Figure 2.11: 建立開機磁片與救援磁片

┌

雖然可以在已安裝的系統中修改分割區，但這只應交由專業人士處理。否則，稍有錯誤，遺失資料的風險會很高。如果您重新分割使用中的硬碟，請隨後立即重新啟動系統。使用救援系統比在執行時重新分割系統還安全。

└

所有連線硬碟上的現有或建議分割區都顯示在進階對話方塊的清單中。全部硬碟都列示為不帶有數字的裝置，像是 `/dev/hda` 或 `/dev/sda`。各分割區則列示為這些裝置的一部分，像是 `/dev/hda1` 或 `/dev/sda1`。硬碟的大小、類型、檔案系統和裝載點以及他們的分割區也都會顯示在其中。裝載點描述分割區在Linux 檔案系統樹狀結構上的裝載位置。

安裝期間若執行專家對話方塊，同時會自動選取並列出所有可用的硬碟空間。若要提供更多磁碟空間給SUSE LINUX，請由清單的底部開始往上釋放所需的空間(從硬碟的最後一個分割區開始至第一個)。例如，如果您有三個分割區，您不可以把第二個供給SUSE LINUX 專用，而將第三和第一個保留給其他作業系統。

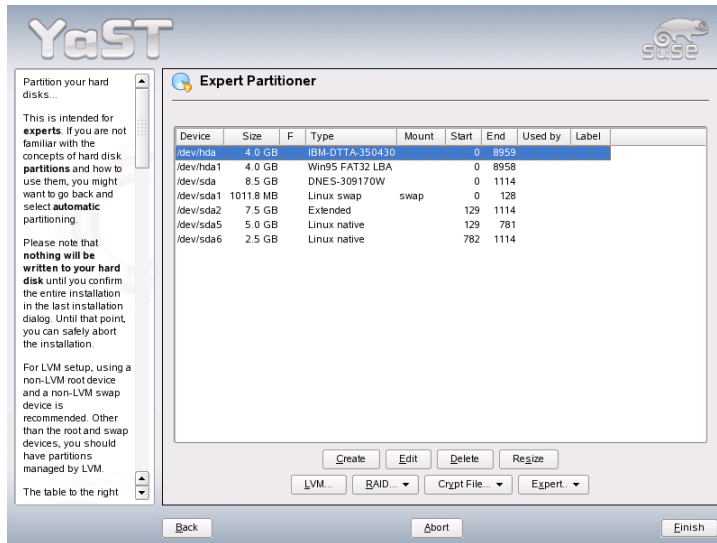


Figure 2.12: YaST 專家分割程式

建立分割區

選擇‘建立’。如果有數個連線的硬碟，會出現一個選擇對話方塊，您可在其中為新分割區選擇一個硬碟。然後，指定分割區類型(主分割區或延伸分割區)。最多可建立四個主分割區，或是三個主分割區和一個延伸分割區。在延伸分割區內建立數個邏輯分割區(請參閱節1.5.4, “分割區類型”)。

選擇要使用的檔案系統，需要時還可選擇一個裝載點。YaST 建議為每個建立的分區都準備一個裝載點。參數的詳細資訊請見下一小節。選擇‘確定’即可套用您所做的變更。新的分割區就會列示在分割區表中。按一下‘下一步’，會採用目前的值。接著在安裝期間，您會回到建議畫面。

分割參數

如果您建立一個新的分割區或修正一個現有分割區，有多項參數可以設定。對於新的分割區，YaST 會設定適當的參數，所以通常並不需要任何修正。若要執行手動設定，請繼續下列步驟：

1. 選擇分割區。

2. '編輯' 分割區並設定參數：

檔案系統ID 即使您在此階段不想要格式化分割區，也請指定一個檔案系統ID給它，以確保分割區的登錄正確。可能的值包括'Linux'、'Linux swap'、'Linux LVM' 和'Linux RAID'。有關LVM和RAID的詳細資訊，請參閱節3.7, "LVM 組態" 和節3.8, "軟體RAID 組態"。

檔案系統 若要在安裝範圍內立即格式化分割區，請為分割區指定下列檔案系統之一：'Swap'、'Ext2'、'Ext3'、'ReiserFS' 或'JFS'。如需各種檔案系統的詳細資訊，請參閱章20, Linux 的檔案系統。

Swap 是一種特殊格式，允許分割區做為虛擬記憶體。ReiserFS 是Linux 分割區的預設檔案系統。ReiserFS、JFS 和Ext3 是日誌檔案系統。這些檔案系統可以在系統毀損後迅速復原系統，因為作業時都已將寫入程序記錄下來。此外，ReiserFS 處理大量小型檔案的速度非常快。Ext2 不是日誌檔案系統。但它非常堅實，並適用於較小的分割區，因為它的管理不需要很大的磁碟空間。

檔案系統選項 在此為選擇的檔案系統設定各種參數。針對使用的各種檔案系統，為進階使用者提供多種選項。

加密檔案系統 如果您啓用加密，所有資料都會以加密格式寫入硬碟。這可增加敏感資料的安全性，但會略為降低系統速度，因為加密處理需要一點時間。有關檔案系統加密的詳細資訊，請參閱節34.3, "加密分割區和檔案"。

fstab 選項 在此為檔案系統(/etc/fstab) 的管理檔案指定各種參數。

裝載點 指定分割區裝載於檔案系統樹狀結構的目錄位置。由各種YaST 提案選擇，或指定任何其他名稱。

3. 選擇'下一步' 來啓用分割區。

如果您是手動分割，請建立一個至少256 MB 的交換分割區。交換分割區可用來釋放目前未使用的資料主要記憶體。這讓使用最頻繁的重要資料可以使用主要記憶體。

進階使用者選項

'進階使用者' 會開啓包含以下指令的功能表：

重新讀取分割區表 重新從磁碟讀取磁碟分割。例如，在文字主控台中手動磁碟分割時需要使用此選項。

刪除分割區表和磁碟標籤 這樣會完全覆寫舊的分割區表。例如，如果使用不常見的磁碟標籤時出現問題，這個指令很有用。使用此方法，會遺失硬碟上所有資料。

更多分割秘訣

如果分割是由YaST執行，而且系統中偵測到其他分割區，這些分割區也都會輸入檔案`/etc/fstab`，便於存取這項資料。這個檔案包含系統中的所有分割區及其屬性，像是檔案系統、裝載點和使用者權限。

Example 2.1: `/etc/fstab`：分割區資料

```
/dev/sda1    /data1    auto      noauto,user 0 0
/dev/sda5    /data2    auto      noauto,user 0 0
/dev/sda6    /data3    auto      noauto,user 0 0
```

無論是Linux或FAT分割區都帶有指定選項`noauto`和`user`。這讓任何使用者都可依需要裝載或解除裝載這些分割區。為了安全理由，YaST不會自動輸入`exec`選項，該選項是執行程式時所必需。不過，若要由此執行程式，您可以手動輸入這個選項。如果您遇到系統訊息，例如解譯器不良或權限遭拒，就需要使用這個方法。

製作分割與LVM

在專家分割程式中，使用‘LVM’存取LVM組態(請參閱節3.7, “LVM組態”)。不過，如果使用的LVM組態已經存在於系統上，只要您第一次在此區段中輸入LVM組態，就會自動啓用。在此情況下，磁碟的分割區若是屬於啓動中的卷冊群組，就無法重新分割，因為當磁碟分割區正在使用時，Linux核心就無法重新讀取這個硬碟的修改分割區表格。不過，如果系統已經有可以運作的LVM組態，就不需要進行實體重新分割。而是變更邏輯卷冊的組態即可。

在實體磁碟區(PV)的開頭，有關磁碟區的資訊會寫入分割區。使用此方式，PV會「知道」它所屬的磁碟區群組。若要為了其他非LVM的使用，而要重新使用這類分割區，最好將此卷冊的開頭刪除。例如，在VG system與PV/dev/sda2中，使用`dd if=/dev/zero of=/dev/sda2 bs=512 count=1`指令，即可完成這個動作。

七

開機的檔案系統

開機的檔案系統(`root`檔案系統或`boot`)不得儲存在LVM邏輯卷冊中。請另外儲存在一般的實體分割區中。

七

2.7.6 設定檔管理員 (SCPM)

SCPM (系統組態設定檔管理) 模組可讓使用者建立、管理系統組態，並可在組態間切換。這對於可在不同位置(不同網路中)，並可由不同使用者使用的行動電腦特別有幫助。然而，即使是固定不動的電腦，此功能也有所幫助，因為它可以使用不同的硬體元件，或測試組態。如需SCPM 基本要點與處理的詳細資訊，請參閱章15, 系統組態設定檔管理。

2.7.7 Systems Services (RunLevel)

SUSE LINUX 可以在數個runlevel 中運作。根據預設，系統會開機為runlevel 5，這可提供多重使用者模式、網路存取與圖形使用者介面(X Window 系統)。其他的runlevel 能提供多重使用者模式與網路功能，但不具備X (runlevel 3)、多重使用者模式但無網路功能(runlevel 2)、單一使用者模式(runlevel 1 與S)、系統暫停(runlevel 0) 以及系統開機(runlevel 6)。

在較高的runlevel 中利用特定服務來連線時若發生問題，其他的runlevel 就會很有幫助。在此狀況下，就可以使用較低的runlevel 將系統開機，以修復服務。許多伺服器都是在沒有圖形使用者介面的狀態下運作，而必須以不具備X 的runlevel 來開機，例如runlevel 3。

通常您只需要標準的runlevel (5)。不過，如果圖形使用者介面停止運作，您可以利用`Ctrl+Alt+F1` 切換到文字主控台，以root 的身分登入，然後以`init 3` 指令切換到runlevel 3 來重新啓動X Window 系統。這會關閉X 視窗系統，留在文字主控台模式。若要重新啓動圖形系統，請輸入`init 5`。

如需SUSE LINUX 中runlevel 的詳細資訊與YaST runlevel 編輯器的說明，請參閱章7, 啓動及設定Linux 系統。

2.7.8 Sysconfig 編輯器

目錄`/etc/sysconfig` 包含的檔案具有SUSE LINUX 最重要的設定。`sysconfig` 會以井然有序的表格顯示所有設定。使用者可以修改值，並儲存到個別的組態檔案中。一般而言，不需要進行手動編輯，因為在安裝套件或設定服務時，就會自動調整檔案。如需`/etc/sysconfig` 與YaST `sysconfig` 編輯器的詳細資訊，請參閱章7, 啓動及設定Linux 系統。

2.7.9 時區選擇

時區已在安裝期間設定，但您可在此進行變更。在清單中按一下您的國家或地區，然後選擇‘當地時間’或‘UTC’ (國際標準時間，以前稱為格林威治標準時

間)。Linux 系統中通常是使用‘UTC’。使用其他作業系統的電腦，如Microsoft Windows，大多數會使用本地時間。

2.7.10 語言選擇

請在此選擇Linux 系統的語言。在YaST 中選擇的語言會套用到整個系統，包括YaST 與桌面環境。

2.8 其他

2.8.1 提交支援要求

購買SUSE LINUX 之後，您有權要求免費的安裝支援。如需支援範圍、地址與電話號碼的詳細資訊，請瀏覽<http://www.novell.com/linux/suse/>。

YaST 可以透過電子郵件，將支援要求直接傳送給SUSE 團隊。可是必須先註冊。從輸入所需的資料開始—您的註冊碼位在CD 封套背面。就您的查詢而論，在下列視窗中選擇問題類別，然後提供問題的描述說明。請參閱圖形2.13, “提交支援要求”。也請閱讀YaST 說明文字，了解該如何說明問題，以方便支援團隊協助您。

如果您需要進階的支援，例如有特殊問題，請參閱<http://support.novell.com/linux/> 以取得詳細資訊。

2.8.2 開機記錄

`/var/log/boot.msg` 開機記錄會包含啓動電腦時顯示的畫面訊息。請使用此YaST 模組來檢視記錄，例如，檢查啓動的所有服務與功能是否運作正常。

2.8.3 系統記錄

系統記錄會將電腦的作業記錄到 `/var/log/messages`。核心訊息會根據日期與時間排序而記錄在此。

The screenshot shows the YaST SUSE Support form. On the left, there is a 'Support Module' section with instructions: 'Enter your personal information as completely as possible in this form. This allows us to reach you if, for example, it is not possible to e-mail you. To avoid additional inquiries, review the support key entered.' The main form area is titled 'SUSE Support' and contains the following fields: 'Enter support data' with radio buttons for 'Mr.' and 'Ms.', 'First name:' and 'Last name:' text boxes, 'Company:' text box, 'Street:' text box, 'ZIP:' and 'City:' text boxes, 'State:' and 'Country:' text boxes, 'E-mail:' text box, and 'Support Key:' text box. At the bottom, there are 'Back' and 'Next' buttons.

Figure 2.13: 提交支援要求

2.8.4 載入廠商的驅動程式光碟

使用此模組，從包含SUSE LINUX 可用驅動程式的Linux 驅動程式光碟，自動安裝設備驅動程式。從頭開始安裝SUSE LINUX 時，可在安裝之後，使用此YaST 模組從廠商光碟載入所需的驅動程式。

2.9 文字模式(ncurses) 中執行YaST

本章節主要僅適用於那些未在其系統上執行X 伺服器，且依賴以文字為基礎的安裝工具的系統管理員及進階使用者。它提供了一些基本資訊，說明如何在文字模式(ncurses) 中啓動與操作YaST。

在文字模式中啓動YaST，會先出現YaST 控制中心。請參閱圖形2.14, “文字模式中的YaST 主視窗”。主要視窗包含3 個區域。由粗的白色框線所圍繞的左框架，內含一些不同模組所屬的類別。會以彩色背景來表示作用中的類別。由細的白色框線所圍繞的右框架，內含作用中類別的可用模組的綜覽。下方框架中有‘說明’按鈕與‘離開’按鈕。

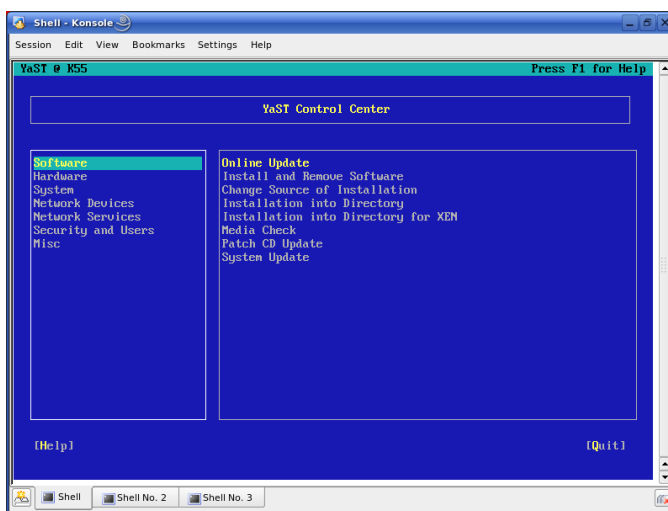


Figure 2.14: 文字模式中的YaST主視窗

啓動YaST控制中心時，會自動選取‘軟體’類別。您可以使用 \downarrow 與 \uparrow 來變更類別。若要啓動所選取的類別中的某個模組，請按 \rightarrow 。此模組選項會加上一個粗的框線。您可以使用 \downarrow 與 \uparrow 以選取想要的模組。您可以按住方向鍵不放，以捲動可用模組清單。選取模組後，模組標題會出現彩色背景，並在下方框架中顯示一段簡要說明。

您可以按 Enter 以啓動想要的模組。模組中的各個按鈕或選項欄位中，都有一個不同顏色的字母(預設為黃色)。您可以使用 $\text{Alt}-(\text{yellow letter})$ 組合鍵來直接選取按鈕，而毋須使用 Tab 來到達想到的地方。您可以按‘離開’按鈕，或在類別綜覽中選取‘離開’然後再按 Enter 鍵，以離開YaST控制中心。

2.9.1 在模組中瀏覽

以下對YaST模組的控制元件所做的說明，假設所有的功能鍵及 Alt 組合鍵都能作用，且未被指定不同的全域功能。如需有關可能的例外狀況的資訊，請參閱節2.9.2, “組合鍵的限制”。

在按鈕與選項清單中瀏覽 使用 Tab 與 $\text{Alt}-(\text{Tab})$ 或 $\text{Shift}-(\text{Tab})$ 以便在含有選項清單的按鈕與框架中進行瀏覽。

在選項清單中瀏覽 在作用中且包含選項清單的框架中，您可以使用方向鍵(↑)和(↓)以便在其中的個別元件之間進行瀏覽。如果框架中個別項目超出其寬度，您可以使用(Shift)+(→)或(Shift)+(←)以水平方式向右捲動或向左捲動。或者使用(Ctrl)+(E)或(Ctrl)+(A)。如果使用(→)或(←)，會導致作用中的框架或目前的選項清單變更，如同在控制中心內一樣，則您也可以使用此組合鍵。

按鈕、圓形按鈕以及核取方塊 如果要選取有空白方括號(核取方塊)的按鈕，或是有空白括號(選項圓鈕)的按鈕，請按(空格)鍵或(Enter)鍵。或者，直接使用(Alt)+(yellow_letter)來選取選項圓鈕與核取方塊。在此狀況中，您不需按(Enter)來做確認。如果您使用(Tab)瀏覽至某個項目，按(Enter)即可執行所選取的動作或啓動個別的功能表項目。

功能鍵 F 鍵(F1)到(F12)可用來快速存取不同的按鈕。因為不同的模組提供不同的按鈕設定(詳細資訊、資訊、新增、刪除等等)，所以各個功能鍵所實際對應的按鈕是依照作用中的YaST 模組而定。您可以使用(F10)當作‘確定’、‘下一步’以及‘完成’來使用。您可以按(F1)來存取YaST 說明，該說明會顯示個別F 鍵所對應的功能。

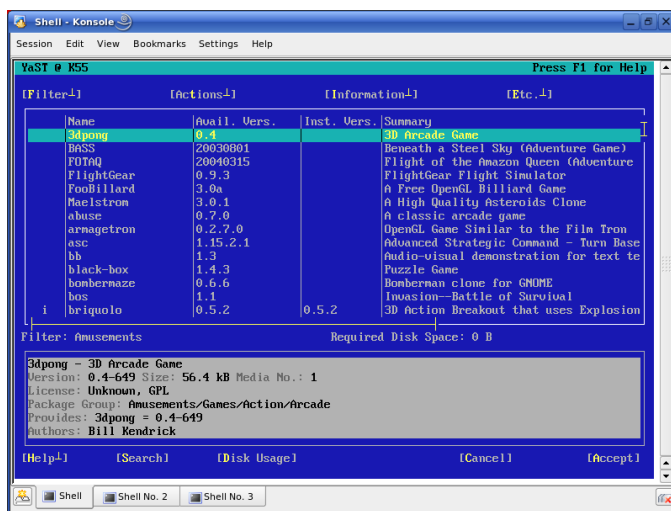


Figure 2.15: 軟體安裝模組

2.9.2 組合鍵的限制

如果您的視窗管理員使用了全部的(Alt)組合，則YaST中的(Alt)組合可能無效。像是(Alt)或(Shift)等鍵也可能事先已由終端機的設定所佔用。

以(Esc)取代(Alt) 您可以使用(Esc)來取代(Alt)，以執行(Alt)捷徑。例如，(Esc)-(H)可取代(Alt)-(H)。

您可以使用(Ctrl)-(F)與(Ctrl)-(B)來往前瀏覽和往後瀏覽。

如果(Alt)和(Shift)組合已先由視窗管理員或終端機所佔用，則請使用(Ctrl)-(F)組合(往前)與(Ctrl)-(B)組合(往後)來代替。

功能鍵的限制 有些功能會使用F鍵。有些特定的功能鍵可能已由終端機所佔用，無法供YaST使用。不過，在純文字主控台中，應該都可以使用各種的(Alt)組合鍵與功能鍵。

2.9.3 啓動個別模組

如果要節省時間，您可以直接啓動個別的YaST模組。若要啓動模組，請輸入`yast <module_name>`。例如，使用`yast lan`可啓動網路模組。使用`yast -l`或`yast --list`，則可以檢視一個清單，其中包含您系統中所有可用的模組。

2.9.4 YOU 模組

和其他的YaST模組一樣，您可以用rootYaST身份登入，從指令行啓動YaST線上更新(YOU)模組：

```
yast online_update .url <url>
```

`yast online_update`會啓動對應的模組。您可以使用`url`選項來指定一個能讓YOU取得所有資訊與修補程式的伺服器(本機或網際網路)。如果您在'啓動模組時，並未指定伺服器，則請在YaST對話方塊中選取伺服器或目錄。設定cron工作，以便透過'設定完全自動更新'來進行自動更新。

2.10 從指令行進行線上更新

使用指令行工具`online_update`，就可以自動更新系統(例如，利用程序檔)。例如，您可能希望系統能搜尋特定伺服器上的更新，以及在指定的時間內周期性地下載修補程式與修補程式資訊。不過，您可能不想要自動安裝修補程式。而想要之後再檢閱修補程式，並選取要安裝的修補程式。

若要使用此工具，請先設定執行下列指令的cron 工作：

```
online_update -u <URL> -g <type_specification>
```

`-u` 提出目錄樹的基本URL，而修補程式應該由此處下載。支援的通訊協定有`http`、`ftp`、`smb`、`nfs`、`cd`、`dvd` 與 `dir`。`-g` 可將修補程式下載到本地目錄而不會安裝它們。或者，透過指定下列類型過濾修補程式：安全性、建議使用或選擇性。如果未指定過濾器，`online_update` 會下載所有新的保安全性與建議使用修補程式。

下載的套件可以立即安裝，無需檢閱個別修補程式。`online_update` 會將修補程式儲存在`/var/lib/YaST2/you/mnt` 目錄。若要安裝修補程式，請執行下列指令：

```
online_update -u /var/lib/YaST2/you/mnt/ -i
```

參數`-u` 可指定要安裝修補程式的本地URL。`-i` 可啓動安裝程序。

若要在安裝之前先檢閱下載的修補程式，請啓動YOU 對話方塊：

```
yast online_update .url /var/lib/YaST2/you/mnt/
```

YOU 會啓動並使用包含下載修補程式的本地目錄，而不是網際網路上的遠端目錄。如同在套件管理員中安裝套件一樣，以相同的方式選擇修補程式進行安裝。您可以使用指令行參數來控制「YaST 線上更新」的動作。語法是`online_update [指令行參數]`。可用的參數與功能如下所示：

- `-u URL` 修補程式應該從目錄樹的基本URL 下載。
- `-g` 只下載修補程式。而不進行安裝。
- `-i` 安裝下載的修補程式。不下載。
- `-k` 檢查是否有新的修補程式。

- c 顯示目前的組態。不執行任何動作。
- p 產品 應該下載修補程式的產品。
- v 版本 應該下載修補程式的產品版本。
- a 架構 應該下載修補程式的產品基本架構。
- d 模擬執行。下載修補程式並模擬安裝(系統維持不變；僅供測試之用)。
- n 不檢查下載檔案的簽名。
- s 顯示可用修補程式的清單。
- v 詳細模式。
- D 供進階使用者用以進行疑難排解的除錯模式。

如需online_update 的詳細資訊，請輸入online_update -h。

特別安裝程序

SUSE LINUX 有多種安裝方式。可使用的方式包括圖形化快速安裝，以及可容許各種手動調整的文字介面安裝。以下各節涵蓋各種安裝程序以及各種安裝來源的使用，包括光碟機和NFS。本章還特別說明如何解決安裝期間發生的問題的詳細資訊並另闢一節說明磁碟分割。

3.1	linuxrc	72
3.2	VNC 的安裝	73
3.3	使用YaST 進行以文字為基礎的安裝	74
3.4	啓動SUSE LINUX	75
3.5	提示和秘訣	77
3.6	指定永久性設備檔案名稱至SCSI 設備	80
3.7	LVM 組態	81
3.8	軟體RAID 組態	86

3.1 linuxrc

每台機器都有特殊的BIOS常式，能在啓動時執行，並起始化硬體。在實際的開機程序期間，這些常式會載入機器要執行的影像檔，而且必須控制下列開機程序。影像一般而言是開機管理員，可讓使用者選取已安裝的系統或是安裝系統。在選取SUSE LINUX的安裝時，會載入含有一個核心與linuxrc程式的開機影像檔。

linuxrc是分析和起始化實際安裝程序系統的程式。根據預設，它是以無使用者互動的方式執行，並會在完成後啓動YaST。如果您需要提供特殊的參數給模組，或是如果硬體偵測失敗，您可能需要啓動手動安裝，以互動方式執行linuxrc。

linuxrc的用途並不僅限於安裝。您也可以用它做為安裝系統的開機工具，甚至可以做為獨立的RAM磁碟救援系統。如需詳細資訊，請參閱節5.4, “SUSE救援系統”。

如果系統使用起始RAM磁碟(initrd)，也稱為linuxrc的外圍程序程序檔，會在開機時載入模組。此程序檔是由/sbin/mkinitrd程序檔動態產生。它完全和用以安裝的linuxrc程式不同，而且不應該與linuxrc混淆。

3.1.1 傳送參數到linuxrc

您也可以將參數傳送至linuxrc，以變更啓動的行為。linuxrc會在磁片上或在/info的initrd中尋找資訊檔。之後，linuxrc會在核心提示時載入參數。您可以編輯/linuxrc.config檔案中的預設值。不過，建議的方法是在資訊檔中變更。

它有可能在手動模式下執行linuxrc。若要這麼做，請在安裝提示下使用“manual=1”參數。

資訊檔案是由關鍵字與格式為key:value的值所組成。您可以在由安裝媒體所提供的開機提示中，輸入多組的關鍵字與值，使用的格式為key=value。所有關鍵字的清單可以在/usr/share/doc/packages/linuxrc/linuxrc.html檔案中取得。下列清單顯示一些具有範例值的最重要關鍵字：

Install:URL (nfs、ftp、hd 等等) 將安裝來源指定為URL。可能的協定有cd、hd、nfs、smb、ftp、http及tftp。URL語法會符合一般網頁瀏覽器所使用的格式，例如：

- `nfs://<server>/<directory>`
- `ftp://[user[:password]@]<server>/<directory>`

Netdevice: `eth0` 如果安裝的主機上有許多乙太網路介面的話，`Netdevice:` 如果在安裝主機上有提供數個乙太網路介面，關鍵字會指定 `linuxrc` 應該使用的介面。

HostIP: `10.10.0.2` 會指定主機的 IP 位址。

Gateway: `10.10.0.128` 如果安裝伺服器並非位於主機的子網路的話，則此關鍵字會指定要抵達安裝伺服器所需經過的閘道。

Proxy: `10.10.0.1` `Proxy:` 關鍵字會定義 FTP 與 HTTP 協定的 proxy。

ProxyPort: `3128` 如果 Proxy 沒有使用預設的埠，則此關鍵字會指定 Proxy 所使用的埠。

Textmode: `0—1` 此關鍵字允許在文字模式中啓動 YaST。

VNC: `0—1` `VNC` 參數會透過 VNC 來控制安裝程序，這會讓沒有圖形主控台的主機更便於安裝。如果開啓，則會在安裝主機上啓動對應的服務。也請參閱 `VNCPassword` 關鍵字。

VNCPassword: `password` 這會為 VNC 安裝設定一組密碼，來控制對工作階段的存取。

UseSSH: `0—1` 此關鍵字讓您可以在文字模式中使用 YaST 執行安裝時，透過 SSH 存取 `linuxrc`。

SSHPassword: `password` 這會設定 `root` 使用者的密碼以存取 `linuxrc`。

Insmod: 模組參數 這會指定核心應該載入的模組及任何該模組所需的參數。模組參數必須以空格來分隔。

AddSwap: `0—3—/dev/hda5` 如果設為 0，系統將不會嘗試啓動交換分割區。如果設為正數，會將對應之數的分割區啓動為交換分割區。或是，指定分割區的完整設備名稱。

3.2 VNC 的安裝

VNC (*virtual network computing*) 是一種用戶端解決方案，可允許遠端 X 伺服器透過輕量型且使用方便的用戶端來存取。此用戶端適用於各種作業系統，包括 Microsoft Windows、Apple 的 MacOS 以及 Linux。

VNC 用戶端(vncviewer) 是用來確定安裝程序中，YaST 的圖形顯示和處理。啓動系統來安裝之前，請準備遠端電腦，讓它可以存取系統透過網路安裝。

3.2.1 準備安裝VNC

要執行VNC 安裝，請傳送特定參數至核心。這個動作必須在核心啓動前完成。要這樣做，在啓動提示輸入以下指令：

```
vnc=1 vncpassword=<xyz> install=<source>
```

vnc=1 指示VNC 伺服器要在安裝系統上啓動。vncpassword 是之後使用的密碼。安裝來源(安裝) 可以手動指定(輸入通訊協定和相關目錄的URL)，也可以包含指示slp:/。在後者案例中，SLP 查詢會自動判斷安裝來源。章23, 網路中的SLP 服務提供SLP 的相關資訊。

3.2.2 要安裝VNC 的用戶端

連接到正在執行VNC 伺服器的安裝電腦，是透過VNC 用戶端所建立。在SUSE LINUX 下，請使用vncviewer。這是xorg-x11-Xvnc 套件的一部份。若要建立從Windows 用戶端到安裝系統的連線，請在Windows 系統上安裝tightvnc 程式。此程式位於第一張SUSE LINUX CD 的/dosutils/tightvnc 目錄中。

啓動您選擇的VNC 用戶端。然後，在提示時，輸入安裝系統的IP 位址，然後是VNC 密碼。

另一種方法是，使用具有Java 功能的瀏覽器來建立VNC 連線。要這樣做，請在瀏覽器的網址欄位輸入：

```
http://<安裝系統的 IP 位址>:5801/
```

建立連線之後，YaST 會啓動而且開始安裝。

3.3 使用YaST 進行以文字爲基礎的安裝

除了透過圖形介面協助安裝之外，還可以用YaST 的文字版本協助安裝SUSE LINUX (主控台模式)。所有YaST 模組也可以在此文字模式中使用。如果您不需要圖形介面(例如同伺服器系統) 或者X Windows 系統不支援您的圖形介面卡時，

文字模式特別有用。使用此安裝模式，視覺障礙的使用者可以透過適當輸出裝置的協助，進行SUSE LINUX 的安裝。

首先，在BIOS 設定啓動順序，從CD-ROM 光碟機啓動。將DVD 或CD 1 放入光碟機，然後重新啓動機器。幾秒鐘之後便會顯示啓動畫面。

使用Ⓜ和Ⓨ，在10 秒內選取‘手動安裝’，防止安裝的系統自動啓動。如果您的硬體需要特殊參數(通常不需要)，在開機選項輸入參數。如果您選擇的鍵盤語言與安裝語言相同，將可正確配置鍵盤。這有利於參數的輸入。

使用(F2) (‘視訊模式’)，設定安裝的螢幕解析度。如果您認為圖形介面卡會在安裝時發生問題，選取‘文字模式’。然後按(Enter)。顯示進度的方塊會顯示載入Linux 核心。核心會先啓動，然後再啓動linuxrc。使用linuxrc 的功能表繼續安裝。

其他開機問題通常可以使用核心參數解決。如果DMA 引發問題，使用開始選項‘安裝—安全設定’。如果您在ACPI (進階組態和電源介面) 發生問題，可以使用以下核心參數。

acpi=off 此參數會關閉電腦的所有ACPI 子系統。如果您的電腦根本無法處理ACPI 或者您認為電腦的ACPI 造成問題，此參數會很有幫助。

acpi=oldboot 除了開機需要的部份之外，關閉ACPI 的其它功能。

acpi=force 永遠啓用ACPI，即使電腦的BIOS 出廠日期是在2000 年以前。若沒有使用acpi=off，設定此參數也會啓用ACPI。

pci=noacpi 關閉新ACPI 系統的PCI IRQ 路由選擇。

在此連接中，使用關鍵字“acpi” 搜尋「支援資料庫」的文章，網址是<https://portal.suse.com>。

如果核心載入或者安裝時發生不明錯誤，選取開機功能表的‘記憶體測試’，檢查記憶體。Linux 要求硬體符合最高標準，這表示記憶體及其時脈必須設定正確。「支援資料庫」中，關鍵字“memtest86” 提供更多資訊。若可能的話，請預留整個晚上執行記憶體測試。

3.4 啓動SUSE LINUX

在安裝之後，請決定日常作業啓動Linux 的方式。以下綜覽會介紹各種不同的Linux 啓動方式。最適合的方式則依使用目的而有所不同。

Linux 開機載入器 Linux 開機管理員是啓動系統最多功能且技術最佳的解決方案，像是GRUB (Grand Unified Bootloader) 或LILO (Linux Loader)，它們可以讓您在開機之前從各種作業系統中選取一個作業系統。您可以在安裝期間設定開機載入器，或是在安裝之後透過YaST的協助來設定開機載入器。

開機磁片 您可以透過開機磁片來啓動Linux。只有在您的系統備有軟碟機時，才能使用此方法。可透過YaST來建立開機磁片。請參閱節2.7.3, “建立開機磁片與救援磁片”。

當您在設定其他開機方式時遇到了困難，或是當您想要延緩關於最後開機機制的決策，開機磁片是非常有用的暫時解決方案。對於OS/2或Windows NT來說，開機磁片也是個適用的解決方案。

└

安裝GRUB或LILO後，有多種BIOS會檢查開機磁區結構(MBR)，並錯誤地顯示病毒警告。進入BIOS並修改對應的設定，便可解決此問題。例如，關閉‘病毒防護’。您可以稍後再重新開啓此選項。然而，如果Linux是您唯一使用的作業系統，就不需這麼做。

└

您可在章8, 開機載入器中，取得有關不同開機方式的詳細討論資訊。

3.4.1 圖形化SUSE 畫面

從SUSE LINUX 7.2 開始，如果使用「`vga=jvaluei`」選項作為核心參數，則圖形化SUSE 畫面會顯示在第一個主控台。如果使用YaST來安裝，則會依照選取的解析度與顯示卡來自動啓動此選項。

3.4.2 停用SUSE 畫面

有三種方法可以停用SUSE 畫面：

必要時，隨時都可以停用SUSE 畫面。

可以在指令行中輸入指令`echo 0 >/proc/splash`以停用圖形式畫面。若要再次啓動它，請輸入`echo 1 >/proc/splash`。

預設停用SUSE 畫面 您可以新增核心參數`splash=0`到開機載入器的組態。

章8, 開機載入器能提供更多有關這方面的資訊。但是，如果您想使用文字模式(較早版本的預設值)，則請設定`vga=normal`。

完全停用SUSE畫面 編譯新的核心且關閉‘framebuffer 支援’中的‘使用開機顯示畫面而非開機標幟’選項。

在核心中關閉framebuffer 支援將會自動關閉開機顯示畫面。如果您使用自訂核心來執行，SUSE 將不會為您的系統提供任何支援。

3.5 提示和秘訣

在某些電腦上，沒有可用的CD-ROM 光碟機，只有可開機的軟碟機設備。若要在這樣的系統上安裝，就必須建立開機磁片，然後用它來啓動系統。

您需要已格式化的3.5" HD 磁片，才能從提供的影像建立可開機的磁片。CD 1 的boot 目錄包含一些磁碟影像。透過合適的公用程式，這些影像可以複製到磁片。按照此方式準備的磁片，稱為開機片。

磁片影像也包括SYSLINUX 載入器與linuxrc 程式。SYSLINUX 允許在開機程序選取核心，並可指定硬體需要使用的任何參數。linuxrc 程式可以為您的硬體載入核心模組，然後開始安裝。

3.5.1 使用rawwritewin 建立開機片

在Windows 中，開機片可以使用圖形公用程式rawwritewin 建立。在CD 1 的目錄dosutils/rawwritewin 可以找到此公用程式。

啓動時，指定影像檔。影像檔位於CD 1 的boot 目錄。您至少需要bootdisk 及modules1 影像。要在檔案瀏覽器列示這些影像，將檔案類型設定成所有檔案。然後在軟碟機插入磁片，按‘寫入’。

可以用相同方法建立其他磁碟影像(modules1、modules2、modules3 和modules4)。如果安裝時，您要解決USB 或SCSI 裝置、網路或PCMCIA 網路卡，則需要這些磁片。如果安裝時使用特殊檔案系統，也需要模組磁片。

3.5.2 使用rawrite 建立開機片

DOS 公用程式rawrite.exe (CD 1，目錄dosutils/rawrite) 可以用來建立SUSE 開機和模組磁片。要使用此公用程式，需要安裝DOS (例如FreeDOS) 或Window 的電腦。

在Windows XP，按照以下方式執行：

1. 插入SUSE LINUX CD 1。
2. 開啓DOS 視窗(在「開始」功能表選取'附屬應用程式'→'命令提示字元')。
3. 執行rawrite.exe 並指定CD 光碟機的正確路徑。範例假設您在硬碟C: 有一個目錄Windows 而且您的CD 光碟機是D:。

```
d:\dosutils\rawrite\rawrite
```

4. 在啓動時，公用程式會詢問要複製的檔案，其來源和目的地。開機磁片的影像位在CD 1 的boot 目錄。檔案名稱是bootdisk。記得指定CD 光碟機的路徑。

```
d:\dosutils\rawrite\rawrite
RaWrite 1.2 - 將磁碟檔案寫入 raw 磁片
```

輸入來源檔名稱：d:\boot\bootdisk 輸入目的地磁碟：a:

輸入目的地磁碟機a: 之後，rawrite 會提示您插入格式化磁片，然後按(Enter)。接下來，顯示複製動作的進度。可以使用(Ctrl)-C 終止程序。要建立多張磁片，重複相同的程序。

3.5.3 在UNIX 類型系統建立開機片

在UNIX 或Linux 系統，您需要CD-ROM 光碟機和數張格式化的磁片。按照以下方法，建立開機片：

1. 如果您需要先格式化磁片，請使用：

```
fdformat /dev/fd0u1440
```

此指令也會檢查磁片是否沒有錯誤。如果媒體有錯誤請勿繼續。

2. 將CD 1 插入光碟機中，然後變更至CD 上的boot 目錄：在目前的SUSE 版本，您不再需要裝載CD。

```
cd /media/cdrom/boot
```

3. 使用以下指令，建立開機片：


```
dd if=bootdisk1 of=/dev/fd0 bs=8k
```

4. 請以bootdisk2 與bootdisk3 重複程序。

boot 目錄的README 檔案，提供磁片影像的詳細資訊。使用more 或less 讀取這些檔案。

可以用相同方法建立其他磁碟影像(modules1、modules2、modules3 和modules4)。如果安裝時，您要解決USB 或SCSI 裝置、網路或PCMCIA 網路卡，則需要這些磁片。如果安裝時使用特殊檔案系統，也需要模組磁片。

建立模組磁片非常重要。如何建立模組磁片的詳細資訊可以在/usr/share/doc/packages/yast2-installation/vendor.html 中找到。

3.5.4 從磁片(SYSLINUX) 開機

開機片可用來處理特殊的安裝需求(例如，無法使用CD-ROM 光碟機時)。開機載入器SYSLINUX (syslinux 套件) 會起始開機程序。系統啟動時，SYSLINUX 會執行最小的硬體偵測，主要包括以下步驟：

1. 程式檢查BIOS 是否提供VESA 2.0- 相容的框架緩衝區支援，然後啟動核心。
2. 讀取監視器資料(DDC 資訊)。
3. 開機載入器設定時，會讀取第一個硬碟的第一個區塊(MBR)，將BIOS ID 對應至Linux 設備名稱。程式嘗試透過BIOS 的lba32 功能讀取區塊，判斷BIOS 是否支援這些功能。

如果啟動SYSLINUX 時，您一直按住(Shift)，會略過所有這些步驟。基於疑難排解用途，請插入以下一行文字

```
verbose 1
```

於syslinux.cfg，讓啟動載入器顯示目前執行的動作。

如果機器未從磁片開機，您可能需要將BIOS 的啟動順序變更成A, C, CDROM。

► x86

在x86 系統，CD 2 也可以用來開機。和CD 1 相反，CD 1 使用開機ISO 影像，而CD 2 是透過2.88 MB 磁碟影像的方式啟動。如果您確定可以從CD 啟動，請使用CD 2，不過它無法與CD 1 同時工作(倒退解決方案)。◀

3.5.5 不支援的CD-ROM 光碟機

支援大部份的CD-ROM 光碟機。如果從CD-ROM 光碟機開機發生問題，請以CD-Set 中的CD 2 開機片來開機。

如果系統沒有CD-ROM 光碟機也沒有軟碟機，仍然可以使用USB、FireWire 或SCSI 來連接外接式CD-ROM 以啟動系統。這大部份是依賴BIOS 及使用的硬體之間的互動。如果您遭遇到問題，有時更新BIOS 可能會有幫助。

3.5.6 從網路來源安裝

有時候無法使用CD-ROM 進行標準安裝。例如，因為您的CD-ROM 是舊的專利光碟機，所以不支援。第二台機器(如筆記型電腦)可能根本沒有CD-ROM，僅有一張乙太網路卡。SUSE LINUX 可透過網路連線執行安裝，不需要CD-ROM。通常這是透過乙太網路，使用NFS 或FTP 完成的。

此方法沒有安裝支援。因此，限有經驗的使用者嘗試以下程序。

要從網路來源安裝SUSE LINUX，需要兩個步驟：

1. 安裝所需的資料(CD、DVD) 必須可以從當成安裝來源的機器取得。
2. 要安裝的系統必須從磁片、CD 或網路上啟動，而且要設定網路。

安裝來源可以透過各種通訊協定取得，例如NFS 和FTP。如需關於實際安裝的資訊，請參閱節3.1.1, “傳送參數到linuxrc”。

3.6 指定永久性設備檔案名稱至SCSI 設備

啟動系統時，會以較動態的方式，指定設備檔案名稱給SCSI 設備。只要設備的編號或組態不變更，就沒有問題。不過，如果新增SCSI 硬碟，而核心先偵測到新硬碟，之後才偵測到舊硬碟，則會指定新名稱給舊硬碟，導致與裝載表格/etc/fstab 中的項目不相符。

要避免此問題，可以使用系統啟動程序檔boot.scsiddev。使用/sbin/insserv 並在/etc/sysconfig/scsiddev 設定它的參數，啓用此程序檔。程序檔/etc/rc.d/boot.scsiddev 會在啟動程序時處理SCSI 設備的設定，並在/dev/scsi/ 底下輸入永久性設備名稱。然後，/etc/fstab 就可以使用這些名稱。。此外，/etc/scsi.alias 可以用來定義SCSI 組態的永久性名稱。在/etc/scsi 中設備的名稱網要在man scsiddev 中有所說明。

在runlevel 的進階模式，啟動層次B 的boot.scsiddev。啟動程序時用於產生名稱所需的連結，會建立在/etc/init.d/boot.d。

設備名稱和udev

至於SUSE LINUX，雖然仍支援boot.scsidev，不過建立永久性設備名稱較好的方法，是使用udev 在/dev/by-id/ 建立具永久性名稱的設備節點。

3.7 LVM 組態

本小節簡短地說明在LVM背後的原則，以及它在許多狀況下好用的基本功能。在節3.7.2, “使用YaST 設定LVM 組態” 中，學習如何使用YaST 設定LVM。

⚠

使用LVM 可能會增加風險，如遺失資料。這些危險也包括應用程式當機、電源中斷和錯誤指令。執行LVM 或重新設定磁碟區前，請儲存您的資料。決不要在沒有備份的情形下工作。

⚠

3.7.1 邏輯磁碟區管理

邏輯磁碟區管理(Logical Volume Manager, LVM) 可以在數個檔案系統上彈性地散佈硬碟空間。在安裝過程中的啓始分割已經完成後，有時需要變更硬碟空間的分割，因此開發此工具。因為要修改執行系統上的分割區很困難，LVM 提供記憶體空間的虛擬集區(磁碟區群組，簡稱VG)。如有必要，可從虛擬集區建立邏輯磁碟區(LV)。作業系統可以存取這些LV，而不是存取實體分割區。磁碟區群組可以延伸至一個以上的磁碟，因此數個磁碟或是磁碟的某些部份可能會構成單一的VG。LVM 這種方法提供從實體磁碟空間擷取的方法，允許使用比實體重新分割更為簡單和安全的方式來變更分割。您可以在節1.5.4, “分割區類型” 與節2.7.5, “磁碟分割” 中找到有關實體分割的背景資訊。

圖形3.1, “實體分割與LVM” 比較實體分割(左邊) 與LVM 分割(右邊)。在左邊，單一個磁碟已分割為三個實體分割區(PART)，每一個都會指定裝載點(MP)，讓作業系統存取它們。在右邊，已經個別將兩個磁碟分割成兩個及三個實體分割區。已經定義兩個LVM 磁碟區群組(VG 1 與VG 2)。VG 1 包含DISK 1 的兩個磁碟區以及DISK 2 的一個磁碟區。VG 2 包含DISK 2 其餘的兩個磁碟區。在LVM 中，

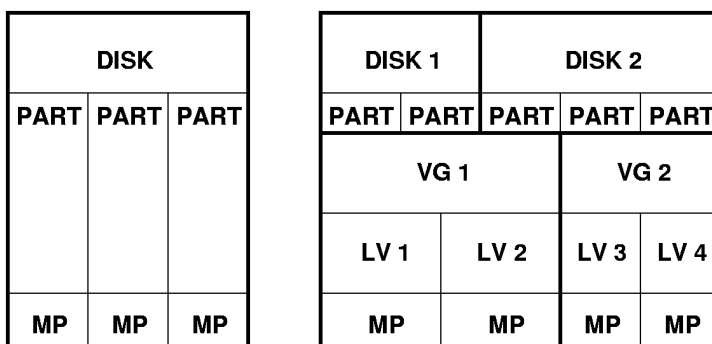


Figure 3.1: 實體分割與LVM

在磁碟區群組中合併的實體磁碟分割區稱為實體磁碟區(PV)。在某些磁碟區群組中，已經定義四個邏輯磁碟區(LV 1 至 LV 4)，作業系統可以透過指定的裝載點來使用。在不同的邏輯磁碟區之間的邊緣，不需要對齊任何分割區的邊緣。請參閱此範例中LV 1 與LV 2 之間的邊緣。

LVM 功能：

- 數個硬碟或分割區可以結合成一個大的邏輯磁碟區。
- 如果組態適用，當可用空間耗盡時，可以擴大LV (如 /usr)。
- 使用LVM，甚至可以在執行的系統中新增硬碟或LV。然而，這種作法需要能執行此動作的熱交換式硬體。
- 可以啓動「等量分割模式」，將邏輯磁碟區的資料流分散至數個實體磁碟區。如果這些實體磁碟區是在不同的磁碟上，這可改善讀寫效能，就像RAID 0 一樣。
- 快照功能能夠讓執行系統中的備份(特別是伺服器) 成為一致。

使用LVM 的這些功能，對於使用頻繁的家用個人電腦或小型伺服器而言，在效能上可以看到改善。如果您的資料會一直累積，如資料庫、音樂歸檔或使用者目錄等，LVM 就是適合您的工具。這樣能夠允許比實體硬碟還大的檔案系統。LVM 的另一個好處是最大可以增加到256 個LV。不過，請記住使用LVM 與使用傳統分割區是不同的。有關設定LVM 的說明及詳細資訊，請參閱官方網站的LVM HOWTO (<http://tldp.org/HOWTO/LVM-HOWTO/>)。

從核心2.6 版本開始，即可使用LVM 2 版本，它可以向下相容之前的LVM，而且可以繼續管理舊的磁碟區群組。建立新的磁碟區群組時，請決定要使用新的格式或能夠向下相容的版本。LVM 2 不需要任何核心修補程式。它利用整合於核心2.6 中的設備映射程式。此核心僅支援LVM 2 版本，因此，談到LVM 時，本節永遠是指LVM 2 版本。

3.7.2 使用YaST 設定LVM 組態

YaST LVM 組態可以從YaST 專家分割程式完成(請參閱節2.7.5, “磁碟分割”)。這個專業磁碟分割工具讓您編輯和刪除現有磁碟分割，以及建立應該與LVM 一起使用的新磁碟分割。接著，先按一下‘建立’→‘不要格式化’，以建立LVM 分割區，再選取‘0x8E Linux LVM’ 做為分割區的識別碼。在建立所有要與LVM 一起使用的分割區後，按一下‘LVM’ 以啓動LVM 組態。

建立磁碟區群組

如果在系統上沒有磁碟區群組，將會提示您新增一個磁碟區群組(請參閱圖形3.2, “建立磁碟區群組”)。您可以使用‘新增群組’ 來建立其他的群組，但是通常單一磁碟區群組就已足夠。建議以system 作為系統檔案所在之磁碟區群組的名稱。實體擴充大小定義了磁碟區群組中實體區塊的大小。在磁碟區群組中的所有磁碟空間都會以此大小的區塊來處理。此值通常設為4 MB，並允許將實體及邏輯磁碟區的最大容量設為256 GB。只有在需要大於256 GB 的邏輯磁碟區時，才需要增加實體擴充大小的容量(例如，設為8、16 或32 MB)。

設定實體磁碟區

一旦建立磁碟區群組，以下對話方塊就會列出具有“Linux LVM” 或“Linux native” 類型的所有分割區。不會顯示交換和DOS 分割區。如果已經指定分割區給磁碟區群組，磁碟區群組的名稱就會顯示在清單中。未指定的分割區以“-” 表示。

如果有數個磁碟區群組，請在左上角的選擇方塊中設定目前的磁碟區群組。右上角的按鈕可以建立其他的磁碟區群組以及刪除現有的磁碟區群組。僅能刪除沒有指定分割區的磁碟區群組。所有指定給磁碟區群組的分割區，又稱為實體磁碟區(PV)。

若要新增之前未指定的分割區給選取的磁碟區群組，請先按一下分割區，再按‘新增磁碟區’。此時，磁碟區群組的名稱，會輸入到選取分割區的旁邊。指定為LVM 保留的所有分割區給磁碟區群組。否則，仍然不會使用分割區上的空間。結束此對話方塊前，每個磁碟區群組必須指定至少一個實體磁碟區。在指定所有的實體磁碟區後，按一下‘下一步’ 以繼續邏輯磁碟區的組態。

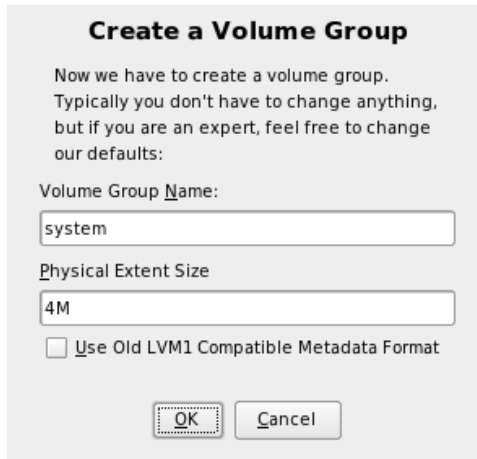


Figure 3.2: 建立磁碟區群組

設定邏輯磁碟區

在磁碟區群組已經使用實體磁碟區填滿後，在下一個對話方塊中定義作業系統應該使用的邏輯磁碟區。在左上角的選項方塊中修改目前的磁碟區群組。接下來，會顯示目前磁碟區群組的可用空間。下方的清單包含該磁碟區群組中所有的邏輯磁碟區。指定裝載點的所有標準Linux分割區、所有交換分割區、以及所有已經存在的邏輯磁碟區都列示於此。視需要‘新增’、‘編輯’以及‘移除’邏輯磁碟區，直到在磁碟區群組中的所有空間都使用完畢。至少指定一個邏輯磁碟區給每個磁碟區群組。

若要建立新的邏輯磁碟區，請按一下‘新增’，然後填寫開啓的快顯視窗。可以輸入磁碟分割、大小、檔案系統以及裝載點。一般而言，如reiserfs或ext2之類的檔案系統，是建立於邏輯磁碟區上，接著再指定裝載點。儲存於此邏輯磁碟區上的檔案，可以在已安裝系統的此裝載點上找到。此外也可以將邏輯磁碟區中的資料流分散至數個實體磁碟區(等量分割)。如果這些實體磁碟區是在不同的硬碟上，通常可以改善讀寫效能(像RAID 0一樣)。不過，具有n個等量磁區的等量LV，只有在LV所需的硬碟空間可以平均分散給n個實體磁碟區時，才能正確建立。例如，如果只有兩個可用的實體磁碟區，那麼就不可能建立具有三個等量磁區的邏輯磁碟區。

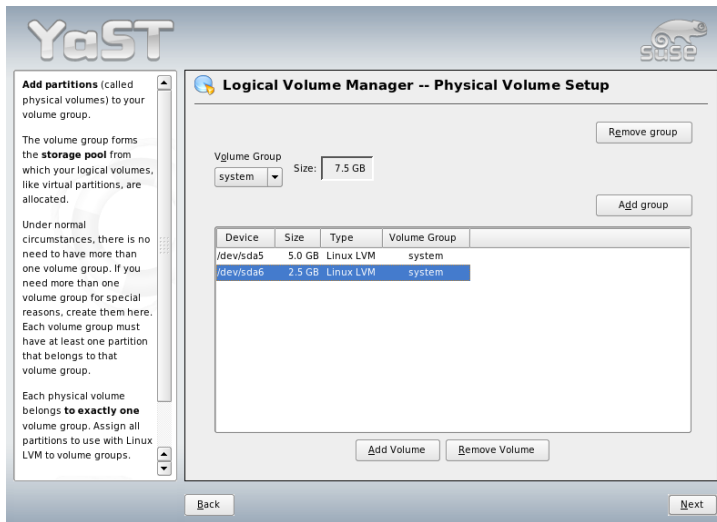


Figure 3.3: 實體磁碟區設定



等量磁區

YaST 在此時沒有機會驗證您所輸入的等量磁區之正確性。在此所犯的錯誤只有稍後在磁碟上執行LVM時才會顯示出來。



如果已經在系統上設定LVM，現在就可以輸入現有的邏輯磁碟區。在繼續前，請指定適當的裝載點到這些邏輯磁碟區。使用‘下一步’，返回「YaST 專家分割程式」，然後在那完成您的工作。

直接LVM 管理

如果您已設定LVM，而且只想變更某些項目，可以使用另一種方式來完成。在「YaST 控制中心」選取‘System’→‘LVM’。基本上此對話方塊允許如上方所描述的相同動作，除了實體分割以外。它以兩個清單顯示現有的實體磁碟區及邏輯磁碟區，而且您可以使用已經描述的方法來管理LVM 系統。

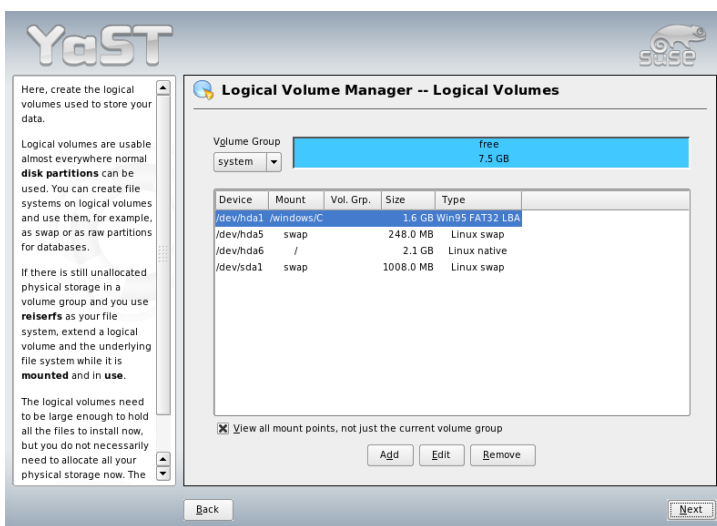


Figure 3.4: 邏輯磁碟區管理

3.8 軟體RAID 組態

RAID (多餘低價磁碟陣列, Redundant Array of Inexpensive Disks) 的用途是將數個硬碟分割區組合一個大型「虛擬」硬碟, 以達最佳化效能、資料安全性或是兩者兼具的功能。但是, 使用此方法, 是犧牲一個好處來換取另一個。大部分RAID 控制器使用SCSI 通訊協定, 因為它可利用比IDE 通訊協定更有效的方式處理較大量的硬碟, 並且更適合指令的平行處理。有部分的RAID 控制器支援IDE 或SATA 硬碟。請參照<http://cdb.suse.de> 的Hardware Database (硬體資料庫)。

3.8.1 軟體RAID

和通常很貴的RAID 控制器一樣, 軟體RAID 也可以承擔這些工作。SUSE LINUX 藉由YaST 的協助, 提供將數個硬碟組合成一個軟體RAID 系統的選項, 這對於硬體RAID 是非常合理的替代品。RAID 一詞是表示將數個硬碟結合成RAID 系統的一些策略, 每個都有不同的目標、優點及特色。這些變化通常稱為RAID 層級。

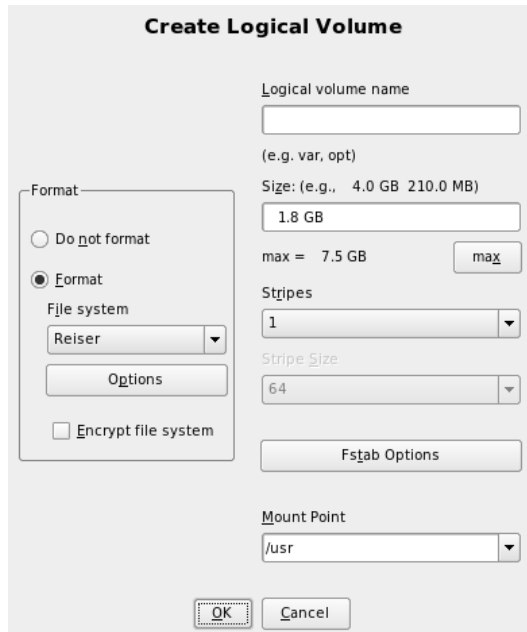


Figure 3.5: 建立邏輯磁碟區

常用RAID 層級為：

RAID 0 此層級將每個檔案的區塊分散於多個磁碟，以提升您的資料存取效能。實際上，它不是真的RAID，因為它不提供資料備份，但是此類型系統的名稱RAID 0已經成為標準。使用RAID 0，就可將兩個以上的磁碟聚集在一起。效能非常好，但是如果其中一個硬碟錯誤，RAID系統便會損毀而且資料會遺失。

RAID 1 此層級對資料提供足夠的安全性，因為資料是以1:1複製到另一個硬碟。這就是所謂的**硬碟鏡射**。如果其中一個硬碟損毀，另一個硬碟上有其內容的複本。只有其中一個會受到損害，但不會危害到資料。在複製程序中的寫入效能比使用單一磁碟存取時較差(慢了百分之十到二十)，但是讀取存取卻較任何一般實體硬碟快得多，因為資料已複製，因此可平行掃描。一般而言，可以說「層級1」比單一磁碟的讀取異動率快了將近兩倍，而且與單一磁碟的寫入異動率幾乎相同。

RAID 2 與 RAID 3 這些都不是一般的 RAID 實作。「層級 2」在是位元層級等量分割資料，而不是在區塊層級。「層級 3」提供具有專用同位磁碟的位元層級等量分割，但是無法同時服務多個要求。這兩個層級都很少使用。

RAID 4 「層級 4」提供與「層級 0」相同的區塊層級等量分割，並且結合專用的同位磁碟。在資料磁碟失敗時，會使用同位資料以建立替代的磁碟。不過，同位磁碟可能造成寫入存取的瓶頸。儘管如此，有時還是會使用「層級 4」。

RAID 5 RAID 5 是在「層級 0」與「層級 1」之間效能和備用方面最佳的折衷方法。硬碟空間等於使用的磁碟數減一。使用 RAID 0 可將資料分布至各個硬碟。在其中一個分割區上建立的同位區塊，即是基於安全性的因素。它們以 XOR 互相連結；在系統失敗時，透過 XOR 藉由對應同位區塊重新建構內容。使用 RAID 5，不會有一個以上的硬碟同時失敗。如果一個硬碟失敗，必須立即更換以避免資料遺失的風險。

其他的 RAID 層級 已經開發 一些 其他的 RAID 層級 (RAIDn、RAID 10、RAID 0+1、RAID 30、RAID 50 等等)，有些是硬體廠商所建立的專用實作。這些層級並不是很普遍，因此在這裏並不做說明。

3.8.2 使用 YaST 進行軟體 RAID 組態

YaST 軟體 RAID 組態可以從 YaST Expert Partitioner 完成 (請參閱節 2.7.5, “磁碟分割”)。這個專業磁碟分割工具讓您編輯和刪除現有磁碟分割，以及建立應該與軟體 RAID 一起使用的新磁碟分割。接著，先按一下‘建立’ → ‘不要格式化’，以建立 RAID 分割區，再選取‘0xFD Linux RAID’ 做為分割區的識別碼。對於 RAID 0 和 RAID 1，至少需要兩個分割區。對於 RAID 1，通常剛好兩個而不需更多。如果使用 RAID 5，至少需要三個分割區。建議您僅使用大小相同的分割區。RAID 分割區應該儲存在不同的硬碟中，以減少當其中一個損壞時 (RAID 1 和 5) 遺失資料的風險，並最佳化 RAID 0 的效能。在建立所有的磁碟區以便和 RAID 一起使用後，按一下‘RAID’ → ‘建立 RAID’ 以啟動 RAID 組態。

在下一個對話方塊中，在 RAID 層級 0、1 及 5 之間選擇 (請參閱節 3.8.1, “軟體 RAID” 以取得詳細資訊)。按一下‘下一步’ 後，下列對話方塊會列出具有“Linux RAID” 或“Linux native” 類型的所有分割區 (請參閱圖形 3.6, “RAID 分割區”)。不會顯示交換和 DOS 分割區。如果已經指定分割區給 RAID 磁碟區，RAID 設備 (例如 /dev/md0) 的名稱就會顯示在清單中。未指定的分割區以“--” 表示。

若要新增之前未指定的分割區給選取的 RAID 磁碟區，請先按一下分割區，再按‘新增’。此時，RAID 設備的名稱，會輸入到選取分割區的旁邊。指定保留

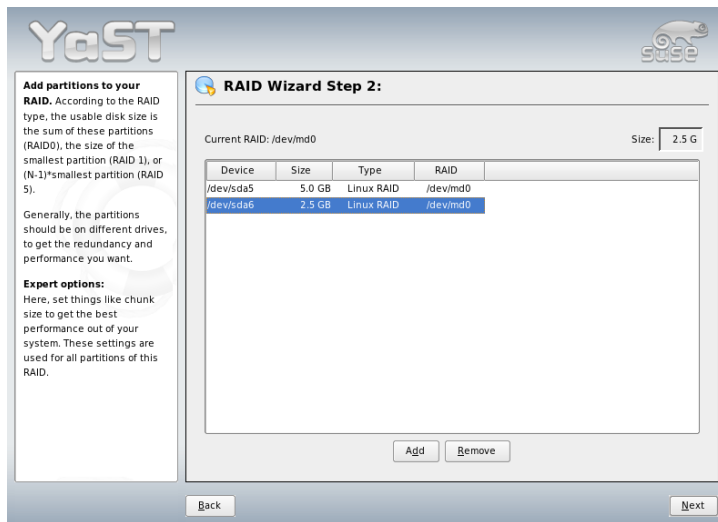


Figure 3.6: RAID 分割區

給RAID的所有分割區。否則，仍然不會使用分割區上的空間。在指定所有的分割區後，按一下‘下一步’以進入設定對話方塊，在此您可以微調效能(請參閱圖形3.7,“檔案系統設定”。)

使用傳統分割時，會設定要使用的檔案系統、加密以及RAID磁碟區的裝載點。檢查‘永久超級區塊’以確定在開機時可識別RAID分割區。在按一下‘完成’以完成組態之後，請參閱/dev/md0設備及其他在專家分割程式中以RAID表示的設備。

3.8.3 疑難排解

檢查檔案`/proc/mdstats`以找出RAID分割區是否已損毀。當系統失敗時，請關閉Linux系統並使用以相同方式磁碟分割的新硬碟來更換損壞的硬碟。然後重新啓動系統，並輸入`mdadm /dev/mdX --add /dev/sdX`指令。當然您必須以特定的設備識別碼取代‘X’。如此可將硬碟自動整合到RAID系統並完整地重新建構。

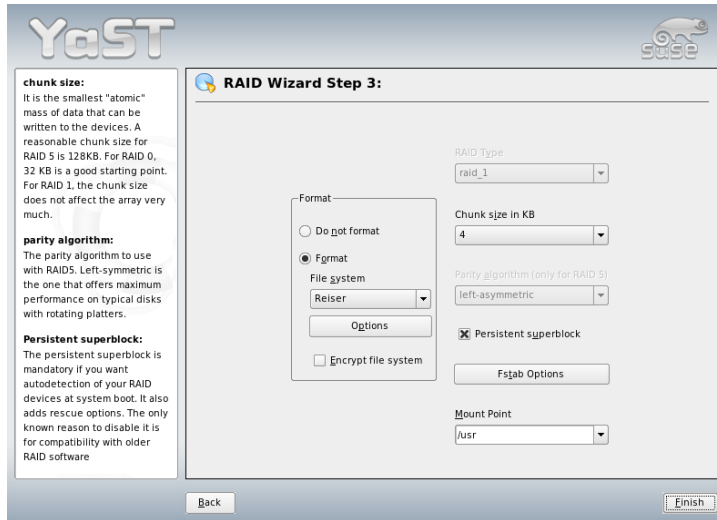


Figure 3.7: 檔案系統設定

3.8.4 若需詳細資訊

可在下列網址的HOWTO 中找到組態指南及軟體RAID 的詳細資訊：

- [/usr/share/doc/packages/raidtools/Software-RAID.HOWTO.html](http://usr/share/doc/packages/raidtools/Software-RAID.HOWTO.html)
- <http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html>

也有Linux RAID 郵寄清單可供參考，如<http://www.mail-archive.com/linux-raid@vger.rutgers.edu>。

更新系統和套件管理

SUSE LINUX 不必完全重新安裝即可更新現有系統。更新的方式有兩種：更新個別的軟體套件和更新整個系統。也可以利用套件管理員RPM 來手動安裝套件。

4.1	更新SUSE LINUX	92
4.2	軟體隨著版本而變更	93
4.3	RPM — 套件管理員	109

4.1 更新SUSE LINUX

軟體通常會隨著版本的更新而「擴增」。因此在更新之前，先使用`df`來檢視可用的分割空間。如果您認為您的磁碟空間可能不夠，請在更新前先確保您資料的安全，然後再進行分割。每個分割區應該多大並沒有常規可循。分割區的大小視您的特定分割設定檔、選定的軟體和SUSE LINUX的版本而定。

請參閱檔案，或者DOS或Windows CD上的`README.DOS`檔。此檔包含本手冊付梓後的所有變更。

4.1.1 準備

更新之前，先將舊的組態檔案複製到其它媒體，例如資料流、可攜式硬碟或ZIP磁碟，以確保資料的安全。此作業主要適用於儲存在`/etc`中的檔案、一些目錄及`/var`和`/opt`中的檔案。您最好也將`/home`（即HOME目錄）中的使用者資料複製到備份媒體。將此資料備份為`root`。只有`root`才可讀取所有本地檔案。

開始更新前，請記住這個`root`分割區。`df /`指令會列出`root`分割區的設備名稱。在範例4.1, “使用`df -h`來列示”中，要記下的`root`分割區為`/dev/hda2`（裝載為`/`）。

Example 4.1: 使用`df -h`來列示

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       1.9G  189M  1.7G  10%  /dos
/dev/hda2       8.9G   7.1G  1.4G  84%  /
/dev/hda5       9.5G   8.3G  829M  92%  /home
```

4.1.2 可能的問題

在`/etc`中檢查密碼和群組

更新系統之前，先確定`/etc/passwd`和`/etc/group`中沒有任何語法錯誤。為達此一目的，應啟動確認公用程式`pwck`和`grpck`做為`root`，並更正回報的錯誤。

PostgreSQL

在更新PostgreSQL (postgres)之前，先傾印資料庫。請參閱手冊中的pg_dump。只有當您在更新前使用PostgreSQL的情況下，才需要執行此一動作。

4.1.3 使用YaST 更新

依照節4.1.1, “準備” 中的準備程序大綱，您現在可以開始更新您的系統了：

1. 依照節1.1, “系統啓動進行安裝” 中說明的安裝方式將您的系統開機。在YaST中，選取一種語言，並選取‘更新現有的系統’。請不要選取‘新的安裝’。
2. YaST 決定是否有多個root 分割區。如果只有一個，繼續進行下一個步驟。如果不只一個，請選取正確的分割區，並以‘下一步’來確認(範例節4.1.1, “準備” 中選擇了/dev/hda2)。YaST 讀取此分割區上原來的fstab 以分析並裝載分割區上所列的檔案系統。
3. 接著您才能在更新時備份系統檔案。此一選項會使更新程序變慢。如果您沒有新近的系統備份檔，則可選擇此項作業。
4. 在下列對話方塊中，選擇只更新已安裝的軟體，或將新的軟體元件加入系統(更新模式)。最好能接受建議的組合，例如‘預設系統’。稍後可使用YaST 進行調整。

4.1.4 更新個別套件

不論您整體的更新環境為何，您都可以更新個別的套件。不過，此後您必須負責確保您系統的一致性。可在<http://www.novell.com/linux/download/updates/> 中找到更新建議。

根據您的需要，從YaST 套件選擇清單中選取元件。如果您選取的套件會影響系統運作，YaST 會發出警告。此種套件只能在更新模式下更新。例如，許多套件都包含共用的程式庫。如果您在運作中的系統下更新這些程式和應用程式，可能會出現故障。

4.2 軟體隨著版本而變更

隨著版本而變更的個別情況詳述如下。如摘要所述，是否已完全重新設定基本設定、是否已將設定檔移至他處，或者是否已大幅變更一般應用程式。有些大幅度

的修改影響到使用者階層或管理者階層的日常系統使用，這些修改在此處均有說明。

各個版本的問題和特殊議題，在識別後已公佈於線上。請參閱下列連結。使用YaST線上更新(YOU)，個別套件的重要更新可在<http://www.novell.com/products/linuxprofessional/downloads/> 找到—請參閱節2.2.3, “YaST線上更新”。

4.2.1 從8.1到8.2

問題及特殊議題：<http://portal.suse.com/sdb/en/2003/04/bugs82.html>

- 以nVidia為基礎的圖形介面卡的3D支援(變更)：不再包含RPM NVIDIA-GLX/NVIDIA_kernel (包括程序檔switch2nvidia_glx)。從nVidia網站(<http://www.nvidia.com>)下載Linux IA32的nVidia安裝程式，安裝此安裝程式中的驅動程式，並利用SaX2或YaST來啓用3D支援。
- 在新安裝的過程中，會安裝xinetd而非inetd並且會以安全值進行設定。請參閱目錄/etc/xinetd.d。不過在系統更新時，會保留inetd。
- 使用的PostgreSQL版本為7.3。如果先前使用的是版本7.2.x，請執行*dump/restore*及*pg_dump*。如果您的應用程式對系統目錄進行查詢，則需要更多的調整，因為版本7.3導入了配置。如需更多資訊，請參閱http://www.ca.postgresql.org/docs/momjian/upgrade_tips_7.3。
- stunnel版本4已不再提供任何指令行選項。不過，附帶的程式檔/usr/sbin/stunnel3_wrapper可將指令行選項轉換為適用於stunnel的組態檔案，並在啓動程式時加以使用(以您的選項來取代「選項」)：

```
/usr/sbin/stunnel3_wrapper stunnel 選項
```

所產生的組態檔案會列印到預設的輸出，如此即可利用這些規格來產生永久的組態檔案。

- openjade (openjade) 為DSSSL引擎，在db2x.sh (docbook-toys) 執行時使用，以取代jade (jade_ds1)。為了提高相容性，不需要前置字元o即可取得個別程式。

如果您的應用程式相依於目錄jade_ds1及先前安裝於此的檔案，請將它們移到新目錄/usr/share/sgml/openjade，或以root身份並使用以下指令來建立連結：


```
cd /usr/share/sgml
rm jade_dsl
ln -s openjade jade_dsl
```

為避免與`rzsz` 相衝，指令行工具`sx` 仍稱為`s2x`、`sgml2xml` 或`osx`。

4.2.2 從8.2 到9.0

問題及特殊議題：<http://portal.suse.com/sdb/en/2003/07/bugs90.html>

- 現在已經可以取得RPM 套件管理員的版本4。用來建立套件的功能已移到其它個別程式`rpmbuild`。`rpm` 仍用於安裝、更新和資料庫查詢。請參閱節4.3, “RPM — 套件管理員”。
- 套件`foomatic-filters` 現在可進行列印。其內容是由`cups-drivers` 分割而來，因為即使未安裝CUPS 仍可用它來列印。如此，YaST 即可支援獨立於列印系統(CUPS、LPRng) 外的組態。此套件的組態檔案為`/etc/foomatic/filter.conf`。
- LPRng 和`lpdfilter` 現在也都需要套件`foomatic-filters` 和`cups-drivers`。
- 使用`/etc/xml/suse-catalog.xml` 中的項目，可存取附帶軟體套件中的XML 資源。不可使用`xmlcatalog` 來編輯此檔案，因為這會刪掉用來正確更新所需的結構性備註。如需存取`/etc/xml/suse-catalog.xml`，請使用`/etc/xml/catalog` 中的`nextCatalog` 陳述，啓用`xmllint` 或`xsltproc` 這一類的XML 工具，以自動尋找本地資源。

4.2.3 從9.0 到9.1

請參閱「SUSE LINUX 9.1 中的已知問題及特殊功能」，此篇文章位於<http://portal.suse.com> 下的SUSE 「支援資料庫」，關鍵字為「特殊功能」。各個SUSE LINUX 版本均有其適用的文章。

升級為核心2.6

SUSE LINUX 如今已完全建立在核心 2.6。前一版 2.4 已無法再使用，因為附帶的應用程式不適用於核心 2.4。請注意下列細節：

- 模組載入是使用 `/etc/modprobe.conf` 檔來進行設定。`/etc/modules.conf` 檔已經過時。YaST 會試著轉換檔案(也請參閱程序檔 `/sbin/generate-modprobe.conf`)。
- 這些模組的字尾為 `.ko`。
- 燒錄 CD 已不再需要用到模組 `ide-scsi`。
- 已從 ALSA 音效模組選項中移除字首 `snd_`。
- `sysfs` 現在使整個 `/proc` 檔案系統更臻完善。
- 電源管理(特別是 ACPI) 比以前更為進步，並且可以使用 YaST 模組來加以設定。

裝置 VFAT 分割區

在裝置 VFAT 分割區時，必須將參數 `code` 變更為 `codepage`。如果您無法裝置 VFAT 分割區，請檢查檔案 `/etc/fstab` 是否含有舊的參數名稱。

待命及暫停 ACPI

核心 2.6 可支援 ACPI 的待命及暫停。此項功能仍在實驗階段，可能不受某些硬體元件的支援。若要使用此功能，您需要 `powersave` 套件。如需此套件的相關資訊，請參閱 `/usr/share/doc/packages/powersave`。在 `kpowersave` 套件中可找到一個圖形前端。

輸入設備

有關輸入設備的變更，請參閱前述中的入口網站文章「SUSE LINUX 9.1 中的已知問題及特殊功能」，此篇文章位於 <http://portal.suse.com> 下的 SUSE「支援資料庫」，關鍵字為「特殊功能」。

「原始POSIX 線串程式庫」和glibc 2.3.x

與NGPT (「新一代的POSIX 線串」) 連結的應用程式與glibc 2.3.x 不相容。所有未附帶SUSE LINUX 的、受影響的應用程式都必須符合linuxthreads 或NPTL (「原始POSIX 線串程式庫」)。NPTL 是較好的選擇，因為它是未來的標準。

如果NPTL 造成問題，則可設定下列環境變數以使用較早的linuxthreads 執行方式(以個別的核心版本編號取代(核心-版本))：

```
LD_ASSUME_KERNEL=核心-版本
```

可以是下列核心版本編號：

2.2.5 (i386, i586)： 沒有浮動堆疊的linuxthreads

2.4.1 (AMD64、i586、i686)： 有浮動堆疊的linuxthread

有關核心和有浮動堆疊的linuxthreads：使用errno、h_errno 和_res 的應用程式必須包含標題檔(errno.h、netdb.h 和resolv.h) 以及#include。那些使用線串取消多線串支援的C++ 程式，必須利用其環境變數LD_ASSUME_KERNEL=2.4.1 來提示使用linuxthreads 程式庫。

原始POSIX 線串程式庫的調整

包含於SUSE LINUX9.1 中的NPTL 為其線串套件。NPTL 與較早的linuxthreads 程式庫為二進位相容。不過，所有linuxthreads 違反POSIX 標準的區域都需要調整NPTL。包含下列各項：訊號處理、getpid 傳回在所有線串中均相同的值，以及線串處理器在未使用vfork 時無法向pthread_atfork 註冊。

網路介面組態

網路介面組態已經變更。之前，在設定一個不存在的介面後即可啓始硬體。如今，系統會搜尋新硬體並立即予以啓始化，來啓用新網路介面的組態。

組態檔案有了新名稱。由於網路介面的名稱是動態產生的，而且hotplug 設備的使用也愈來愈穩定，因此類似eth0、eth1 等名稱已不再適用於組態設定。此外，一些獨特的指定像是MAC 位址或PCI 插槽，已用來命名介面組態。只要介面名稱一出現您就可以使用。仍可能出現像是ifup eth0 或ifdown eth0 之類的指令。

設備組態位於/etc/sysconfig/hardware。這些設備所提供的介面通常位於/etc/sysconfig/network (有不同名稱)。請參閱/usr/share/doc/packages/sysconfig/README 中的詳細說明。

聲音組態

更新之後必須重新設定音效卡。可使用YaST 音效模組來完成此工作。以root 身份輸入yast2 sound。

以頂層領域.local 做為「連結-本地」領域

解析程式庫將頂層領域.local 視為「連結-本地」領域，並將多重廣播DNS 查詢傳送至多重廣播位址224.0.0.251，連接埠5353，以取代一般的DNS 查詢。這是一個不相容的變更。如果領域.local 已用於名稱伺服器組態，請使用別的領域名稱。如需有關多重廣播DNS 的詳細資訊，請參閱<http://www.multicastdns.org>。

全系統UTF-8 編碼

此系統的預設編碼為UTF-8。因此在執行標準安裝時，會以UTF-8 編碼來進行地區設定，例如en_US.UTF-8。如需更多資訊，請參閱<http://www.suse.de/~mfabian/suse-cjk/locales.html>。

將檔名轉換為UTF-8

先前建立檔案系統中的檔案，並未使用UTF-8 編碼來為檔案命名(除非另有規定)。如果這些檔名含有非ASCII 字元，將成為亂碼。若要加以更正，請使用convmv 程序檔來將檔名編碼轉換為UTF-8。

與2001 POSIX 標準相容的外圍程序工具

在預設狀態下，coreutils 套件中的外圍程序工具(tail、chown、head、sort等等)不再與1992 的POSIX 標準相容，而是與2001 的POSIX 標準相容(單一UNIX 規格，版本3 == IEEE Std 1003.1-2001 == ISO/IEC 9945:2002)。使用環境變數可強制執行較舊版本的作業。

```
_POSIX2_VERSION=199209
```

新值為200112，並且用來做為_POSIX2_VERSION 的預設值。可於<http://www.unix.org> 檢視SUS 標準(免費但必須註冊)。

Table 4.1: POSIX 1992 與POSIX 2001 的比較

POSIX 1992	POSIX 2001
chown tux.users	chown tux:users

<code>tail +3</code>	<code>tail -n 3</code>
<code>head -1</code>	<code>head -n 1</code>
<code>sort +3</code>	<code>sort -k 4</code>
<code>nice -10</code>	<code>nice -n 10</code>
<code>split -10</code>	<code>split -l 10</code>

協力廠商軟體可能還無法與新標準相容。因此，請依照上述來設定環境變數。

過時的`/etc/gshadow`

由於下列原因，`/etc/gshadow` 已成為多餘，因此被放棄並移除：

- `glibc` 已不再予以支援。
- 此檔案沒有正式的介面。甚至陰影套裝軟體中也沒有此介面。
- 用來檢查群組密碼的工具大多不支援此檔，並因此加以忽略。

OpenLDAP

由於資料庫的格式已變更，因此需要重新產生資料庫。在更新時，系統會試著自動執行此轉換工作。不過，當然會出現轉換失敗的狀況。

配置檢查已有長足的進步。因此，許多不符合標準的作業，之前還可能用於LDAP 伺服器，如今已不再適用。

為了檢視ACL，組態檔案的語法已做了部分變更。安裝好之後，可在檔案`/usr/share/doc/packages/openldap2/README.update` 中找到有關更新的資訊。

以Apache 2 取代Apache 1.3

Apache 網頁伺服器(版本1.3) 已被Apache 2 取代。有關版本2.0 的詳細文件資訊可在網頁<http://httpd.apache.org/docs-2.0/en/> 中找到。在裝有HTTP 伺服器的系統上，其更新會移除Apache 套件並安裝Apache 2。因此，

該系統必須以YaST 或手動進行調整。在/etc/httpd 中的組態檔案現在已移到/etc/apache2 中。

可選擇線串或程序，以同時處理多個查詢。程序管理已移到一個獨立的模組，即多重處理模組(Multiprocessing Module, MPM)。據此，Apache 2 需要apache2-prefork 套件(建議使用，以提高穩定性) 或apache2-worker 套件。視MPM 而定，Apache 2 會以不同的方式回應查詢。這會影響其效能及模組的使用。這些特點於節30.4, “執行緒” 中均有詳述。

Apache 2 現在已能支援新一代網際網路協定IPv6。

已執行一項機制，讓模組程式設計人員可指定所要的模組載入程序，除去使用者的負擔。通常，模組的執行程序是很重要的，過去是由載入程序來決定。例如，必須先載入一個只允許通過驗證的使用者存取某些資源的模組，以防止無存取許可權的使用者閱讀網頁。

可利用過濾器來處理Apache 所收到的查詢及其回應。

從Samba~2.x 到Samba~3.x

將Samba~2.x 更新為Samba~3.x 之後，就不能再取得winbind 驗證。其它驗證方式則可繼續使用。下列程式已因此被移除：

```
/usr/sbin/wb_auth  
/usr/sbin/wb_ntlmauth  
/usr/sbin/wb_info_group.pl
```

請參閱<http://www.squid-cache.org/Doc/FAQ/FAQ-23.html#ss23.5>。

OpenSSH 更新(版本3.8p1)

gssapi 支援已被gssapi-with-mic 取代，以防止可能的MITM 攻擊。這兩個版本不相容。這表示您無法以較早版本的Kerberos 票證進行驗證，因為已使用其它驗證方式。

SSH 與終端應用程式

從遠端主機(特別是透過SSH、telnet 和RSH) 建立版本9 (含有作用中的UTF-8 的標準組態) 與較早系統(SUSE LINUX 9.0 及較早版本，其UTF-8 依照預設為非作用中或不受支援) 之間的連線時，終端應用程式可能會顯示錯誤字元。

這是由於OpenSSH 未轉送本地設定所致。因此使用了可能不符合遠端終端設定的系統預設值。這會影響YaST 的文字模式以及遠端主機以一般使用者身份(而非root) 所執行的應用程式。以root 的身份啓動的應用程式只有在使用者變更標準地區設定為root 時才會受到影響(只有LC_CTYPE 是依照預設值來設定)。

捨棄libiodbc

FreeRADIUS 的使用者現在必須連線至unixODBC，因為libiodbc 已被捨棄。

/usr/share/xml 中的XML 資源

FHS (請參閱節A, “標準和規格”) 現在需要XML 資源(DTD、樣式表等等) 以安裝於/usr/share/xml 中。因此，某些目錄已經不存在於/usr/share/sgml。如果您發現問題，請修改您的程序檔和makefiles 或使用正式的目錄(特別是/etc/xml/catalog 或/etc/sgml/catalog)。

可移動的媒體與subfs

可移動的媒體現在已與subfs 整合。不再需要以mount 手動裝載媒體。如果要裝載媒體，只要直接在/media 中變更個別設備即可。如果程式正在使用媒體，則無法將媒體退出。

4.2.4 從9.1 到9.2

請參閱「SUSE LINUX 9.1 中的已知問題及特殊功能」，此篇文章位於<http://portal.suse.com> 下的SUSE 「支援資料庫」，關鍵字為「特殊功能」。

在安裝時啓動提議對話方塊中的防火牆

爲了提高安全性，會在提議對話方塊中的安裝結束前啓動附帶的防火牆解決方案SuSEFirewall2。這表示一開始時所有連接埠都會關閉，需要時則可在提議對話方塊中開啓。依照預設，您無法從遠端系統登入。這也會干擾網路瀏覽及多重廣播應用程式，像是SLP 及(網路上的芳鄰)，以及一些遊戲。您可以使用YaST 來微調防火牆的設定。

在安裝服務與設定服務期間，如果需要網路存取，個別YaST 模組會開啓所有內部與外部模組中必要的TCP 與UDP 埠。不需要時，使用者可關閉YaST 模組中的連接埠或指定其它詳細的防火牆設定。

Table 4.2: 重要服務所使用的連接埠

服務	埠
HTTP 伺服器	根據「清單」內容設定防火牆(僅TCP)
郵件(postfix)	smtp 25/TCP

Samba 伺服器	netbios-ns 137/TCP; netbios-dgm 138/TCP; netbios-ssn 139/TCP; microsoft-ds 445/TCP
DHCP 伺服器	bootpc 68/TCP
DNS 伺服器	領域53/TCP；領域53/UDP
DNS 伺服器 埠對應程式	SuSEFirewall2 的埠對應程式的附加特殊支援 sunrpc 111/TCP；sunrpc 111/UDP
NFS 伺服器	nfs 2049/TCP
NFS 伺服器	附加埠對應程式
NIS 伺服器	啓動portmap
TFTP	tftp 69/TCP
CUPS (IPP)	ipp 631/TCP; ipp 631/UDP

KDE 和IPv6 支援

依照預設，KDE 並未啓用IPv6 支援。您可以使用YaST 的 `/etc/sysconfig` 編輯器來啓用它。停用這個功能的原因，是因為不是所有的網際網路服務提供者都支援IPv6 位址，所以可能會造成瀏覽網頁時出現錯誤訊息，以及網頁的顯示出現延遲現象。

YaST 線上更新與「Delta 套件」

YaST 線上更新現在支援特別的RPM 套件，此套件僅會儲存指定基本套件的二進位變更。這項技術大幅地減少了花費較高的CPU 負擔來重新組合最後套件時的下載時間及套件大小。在 `/etc/sysconfig/onlineupdate` 中，您可以設定YOU 是否使用這些「delta 套件」。請參閱 `file:///usr/share/doc/packages/deltarpm/README` 以取得技術細節。

列印系統組態

在安裝結束前(提議對話方塊) 必須在防火牆組態中打開列印系統所需的連接埠。CUPS 需要連接埠631/TCP 和連接埠631/UDP，而且不應將其關閉，以維持正常作業。連接埠515/TCP (用於舊的LPD 協定) 和Samba 所使用的連接埠也都必須開啓，以透過LPD 或SMB 來列印。

變更為X.Org

從XFree86到X.Org的變更是由相容連結來進行，透過該連結可使用舊名稱來存取重要的檔案和指令。

Table 4.3: 指令

XFree86	X.Org
XFree86	Xorg
xf86config	xorgconfig
xf86cfg	xorgcfg

Table 4.4: /var/log 中的記錄檔

XFree86	X.Org
XFree86.0.log	Xorg.0.log
XFree86.0.log.old	Xorg.0.log.old

在變更為X.Org 的過程中，XFree86* 套件會被重新命名為xorg-x11*。

X11 的終端機模擬器

我們已移除一些終端機模擬器，因為它們已不再提供維護，或是已無法在預設環境中使用，尤其是不支援UTF-8。SUSE LINUX 提供標準終端機，像是xterm、KDE 及GNOME 終端機，以及mlterm (X 的多語系終端機模擬器)，這個終端機可以用來取代aterm 與eterm。

powersave 套件中的變更

/etc/sysconfig/powersave 中的組態檔案已變更：

Table 4.5: 分割/etc/sysconfig/powersave 中的組態檔案

舊的	現在已分割為
/etc/sysconfig/powersave/common	common
	cpufreq
	events
	battery

sleep
thermal

/etc/powersave.conf 已經過時。現有的變數已移至表格4.5, “分割/etc/sysconfig/powersave 中的組態檔案”中所列的檔案。如果您變更了/etc/powersave.conf 中的「事件」變數，則現在必須在/etc/sysconfig/powersave/events 中進行相同的變更。

下列睡眠狀態名稱已經變更：

- 暫停(ACPI S4, APM 暫停)
- 待命(ACPI S3, APM 待命)

寫入到：

- 暫停寫入到磁碟(ACPI S4, APM 暫停)
- 暫停寫入到RAM (ACPI S3, APM 暫停)
- 待命(ACPI S1, APM 待命)

OpenOffice.org (OOo)

目錄： OOo 現在安裝於 /usr/lib/ooo-1.1 而非 /opt/OpenOffice.org。使用者設定的預設目錄現在為 ~/.ooo-1.1 而非 ~/OpenOffice.org1.1。

包裝函式： 有一些新的包裝函式可用來啓動OOo 元件。新名稱列於表格4.6, “包裝函式”中。

Table 4.6: 包裝函式

舊的	新的
/usr/X11R6/bin/OOo-calc	/usr/bin/ocalc
/usr/X11R6/bin/OOo-draw	/usr/bin/oodraw
/usr/X11R6/bin/OOo-impress	/usr/bin/ooimpress
/usr/X11R6/bin/OOo-math	/usr/bin/oomath

/usr/X11R6/bin/OOo-padmin	/usr/sbin/oopadmin
/usr/X11R6/bin/OOo-setup	-
/usr/X11R6/bin/OOo-template	/usr/bin/oofromtemplate
/usr/X11R6/bin/OOo-web	/usr/bin/ooweb
/usr/X11R6/bin/OOo-writer	/usr/bin/oowriter
/usr/X11R6/bin/OOo	/usr/bin/ooffice
/usr/X11R6/bin/OOo-wrapper	/usr/bin/ooo-wrapper

包裝函式現在支援選項`--icons-set`可在KDE和GNOME圖示之間切換。下列選項已不再受到支援：`--default-configuration`、`--gui`、`--java-path`、`--skip-check`、`--language` (其語言現在是由地區設定來決定)、`--messages-in-window`和`--quiet`。

KDE 和GNOME 支援： 可在OpenOffice.org-kde 和OpenOffice.org-gnome 套件中找到KDE 和GNOME 的副檔名。

混音器kmix

已預先將混音器kmix 設定為預設項目。高階硬體有其它混音器，像是QAMix/KAMix、envy24control (限ICE1712) 或hdspmixer (限RME Hammerfall)。

DVD 燒錄

過去我們會從cdrecord 套件套用一個修補程式到cdrecord 二進位，以支援燒錄DVD。新安裝的二進位cdrecord-dvd 已改成具備這個修補程式。

dvd+rw-tools 套件的growisofs 程式現在可以燒錄所有的DVD 媒體(DVD+R、DVD-R、DVD+RW、DVD-RW、DVD+RL)。我們建議您使用此程式，取代套用修補程式的cdrecord-dvd。

多核心

的確可以安裝多個核心。這個功能是指，允許管理員透過安裝新核心來從一個核心升級到另一個核心，接著驗證新核心可以如預期般正常運作，然後再解除安裝舊核心。在YaST 不支援這項功能時，使用`rpm -i (package).rpm` 指令便可輕

易地從外圍程序安裝核心或解除安裝核心。若需有關使用指令行來管理套件的資訊，請參閱節4.3, “RPM — 套件管理員”。

預設的開機載入器功能表包含一個核心項目。在安裝多個核心之前，您可以為額外的核心新增一個項目，以方便您選取這些核心。在安裝新核心之前啟動的核心可以 `vmlinuz.previous` 及 `initrd.previous` 來存取。透過建立與預設項目相似的開機載入器項目，並讓這個項目從參照 `vmlinuz` 與 `initrd`，改成參照 `vmlinuz.previous` 與 `initrd.previous`，就可以存取之前啟動的核心。此外，GRUB 與 LILO 支援萬用字元開機載入器項目。請參閱 GRUB 資訊頁面 (`info grub`) 以及 `lilo.conf` (5) 手冊頁，以取得詳細資訊。

4.2.5 從9.2 到9.3

請參閱「SUSE LINUX 9.3 中的已知問題及特殊功能」，此篇文章位於 <http://portal.suse.com> 下的 SUSE 「支援資料庫」，關鍵字為「特殊功能」。

啓動核心提示的手動安裝

開機載入器畫面已不再提供‘手動安裝’模式。在開機提示中使用 `manual=1`，您仍然能夠讓 `linuxrc` 進入手動模式。通常這不是必要的，因為您可以直接在核心提示中設定安裝選項，像是 `textmode=1`，或是設定一個 URL 做為安裝來源。

網路驗證的 Kerberos

Kerberos 取代 `heimdal` 成為預設的網路驗證。無法自動轉換現有的 `heimdal` 組態。系統更新組態檔案的備份時，會建立表格4.7, “備份檔案” 中顯示的項目。

Table 4.7: 備份檔案

舊的檔案	備份檔案
<code>/etc/krb5.conf</code>	<code>/etc/krb5.conf.heimdal</code>
<code>/etc/krb5.keytab</code>	<code>/etc/krb5.keytab.heimdal</code>

用戶端組態 (`/etc/krb5.conf`) 和 `heimdal` 的組態十分相似。若無特別設定，用 `admin_server` 來替換參數 `kpasswd_server` 就可以了。

接管伺服器 (`kdc/kadmind`) 相關資料是不可能的。系統更新之後，仍舊可以在 `/var/heimdal` 中使用舊的 `heimdal` 資料庫；MIT `kerberos` 仍會維護 `/var/lib/kerberos/krb5kdc` 下的資料庫。

X.Org 組態檔案

組態工具SaX2可將X.Org 組態設定寫入/etc/X11/xorg.conf。全新安裝過程中不會建立從XF86Config到xorg.conf的相容連結。

PAM 組態

common-auth Auth 區段的預設PAM 組態

common-account 帳戶區段的預設PAM 組態

common-password 密碼變更的預設PAM 組態

common-session 工作階段管理的預設PAM 組態

您應該在您的應用程式特殊組態檔案中，加入這些預設組態檔案，因為和修改和維護將近四十個在系統中常見的檔案相比，僅修改和維護一個組態檔是要容易多了。若您之後再安裝一個應用程式，此應用程式會繼承已套用的變更，管理員不需要費事去調整其組態。

這個變更很簡單：若您有下列組態檔(大多數的應用程式預設會有這些檔案)：

```
##%PAM-1.0
auth      required          pam_unix2.so
account   required          pam_unix2.so
password  required          pam_pwcheck.so
password  required          pam_unix2.so      use_first_pass use_authtok
#password required          pam_make.so        /var/yp
session   required          pam_unix2.so
```

您可以將它變更成：

```
##%PAM-1.0
auth      include           common-auth
account   include           common-account
password  include           common-password
session   include           common-session
```

4.3 RPM — 套件管理員

在SUSE LINUX 中，使用RPM (Red Hat 套件管理員) 來管理軟體套件。它的主程式為rpm 和rpmbuild。使用者、系統管理員和套件建立者可在威力強大的RPM 資料庫中查詢安裝軟體的詳細資訊。

基本上，rpm 有五種模式：安裝、解除安裝或更新軟體套件；重建RPM 資料庫；查詢RPM 基礎或個別RPM 歸檔；套件的完整性檢查；以及簽名套件。可使用rpmbuild 從原始來源建立可安裝套件。

可安裝的RPM 歸檔以特殊二進位格式包裝。這些歸檔由要安裝的程式檔和rpm 在安裝期間用來設定軟體套件或儲存在RPM 資料庫中以供文件用途的特定中繼資訊所組成。RPM 歸檔的副檔名通常為.rpm。

rpm 可用來管理LSB 相容套件。請參考節A, “標準和規格” 以取得LSB 相關資訊。

對於許多套件，軟體開發所需的元件(程式庫、標頭、包括的檔案等) 已經放在分別的套件中。只有在您想要自行編譯軟體(例如，最新的GNOME 套件) 時，才需要這些開發套件。可由它們的副檔名-devel 加以識別，例如套件alsa-devel、gimp-devel 和kdelibs-devel。

4.3.1 確認套件驗證性

SUSE LINUX RPM 套件具有GnuPG 簽名。包含指紋的金鑰為：

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>  
Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA
```

指令rpm --checksig apache-1.3.12.rpm 可用來確認RPM 套件的簽名，以判斷它是否確實來自SUSE，或是來自其他可信任的設備。從網際網路更新套件時，特別建議使用。SUSE 公用套件簽名金鑰通常位於/root/.gnupg/ 中。在/usr/lib/rpm/gnupg/ 目錄中也額外放置了金鑰，以便讓一般使用者確認RPM 套件的簽名。

4.3.2 管理套件：安裝、更新與解除安裝

RPM 歸檔的安裝通常很簡單：`rpm -i <套件>.rpm`。使用此指令可安裝套件，但是必須滿足其依存度，而且不可與其他套件衝突。錯誤訊息是rpm 要求必須安裝的套件要符合依存度要求。在背景中，RPM 資料庫確定沒有產生衝突，亦即特定檔案僅可屬於一個套件。藉由選擇不同選項，您可以強迫rpm 忽略這些預設，但是僅供進階使用者使用。否則，會有危害系統完整性的風險，而且可能會危害更新系統的能力。

`-U` 或 `--upgrade` 和 `-F` 或 `--freshen` 選項可用來更新套件，例如，`rpm -F <套件>.rpm`。此指令會移除舊版的檔案，並立刻安裝新檔案。兩個版本之間的不同在於 `-U` 會安裝目前不存在於系統中的套件，而 `-F` 僅更新目前安裝的套件。在更新時，rpm 會使用下列策略小心地更新組態檔：

- 如果系統管理員未變更組態檔，rpm 會安裝適當檔案的新版本。系統管理員不需要做任何動作。
- 如果系統管理員在更新之前變更組態檔，rpm 會將變更的檔案以副檔名 `.rpmorig` 或 `.rpmsave` (備份檔案) 儲存，並安裝新套件的版本，但必須在原始安裝檔案與較新版本不同時才行。如果是此狀況，請比較備份檔案 (`.rpmorig` 或 `.rpmsave`) 與新安裝的檔案，然後再對新檔案做一次變更。之後，請確定刪除所有 `.rpmorig` 和 `.rpmsave` 檔案以避免未來更新的問題。
- 如果組態檔已存在，且如果在 `.spec` 檔案中指定了 `noreplace` 標籤，會出現 `.rpmnew` 檔案。

在更新之後，應該在比較 `.rpmsave` 和 `.rpmnew` 之後將它們移除，才不會妨礙未來的更新。如果RPM 資料庫之前無法辨識檔案，會指定 `.rpmorig` 副檔名。

否則，會使用 `.rpmsave`。換言之，`.rpmorig` 是由RPM 的外來格式更新所造成的。`.rpmsave` 是由舊版RPM 更新為新版RPM 所造成的。`.rpmnew` 沒有透露任何關於系統管理員是否曾對組態檔做過任何變更的資訊。可在 `/var/adm/rpmconfigcheck` 找到這些檔案的清單。部分組態檔 (如 `/etc/httpd/httpd.conf`) 不會覆寫以允許後續操作。

`-U` 切換不是只和使用 `-e` 選項解除安裝以及使用 `-i` 選項安裝相同。如果可能，請使用 `-U`。

若要移除套件，請輸入 `rpm -e <套件>`。如果沒有無法解析的依存度，rpm 僅會刪除套件。理論上無法刪除 `Tcl/Tk`，只要其他應用程式還需要它。即使是此狀況，RPM 可從資料庫呼叫以得到協助。如果此種刪除是 (不論何種原因或在尋常的狀況下) 不可行的，即使不存在額外的依存度，使用選項 `--rebuilddb` 來重建RPM 資料庫可能有幫助。

4.3.3 RPM 與修補程式

若要保障系統的操作安全性，必須經常在系統上安裝更新套件。以前，要除去套件中的問題，只能夠更換整個套件。在小檔案中有問題的大型套件很容易產生大量的資料。但是，SUSE RPM 提供在套件中安裝修補程式的功能。

最重要的考量可用pine 當作範例：

修補程式RPM 是否適用於我的系統？

若要檢查，首先查詢安裝的套件版本。以pine 為例，可使用指令

```
rpm -q pine
pine-4.44-188
```

然後檢查修補程式RPM 是否適用於此版本的pine：

```
rpm -qp --basedon pine-4.44-224.i586.patch.rpm
pine = 4.44-188
pine = 4.44-195
pine = 4.44-207
```

此修補程式適用於三種不同版本的pine。範例中也列出安裝的版本，因此可安裝此修補程式。

修補程式會更換哪些檔案？ 受到修補程式影響的檔案可在修補程式RPM 中輕易地看出。rpm 參數-P 讓您選擇特殊修補程式功能。可使用以下指令顯示檔案清單：

```
rpm -qpP1 pine-4.44-224.i586.patch.rpm
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

或者，如果已經安裝修補程式，可使用以下指令：

```
rpm -qP1 pine
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

修補程式RPM 如何安裝於系統中？ 修補程式RPM 可當作一般RPM 使用。唯一的不同是必須已經安裝適合的RPM。

系統中已經安裝哪個修補程式，是用於哪個套件版本？

使用指令 `rpm -qPa` 可顯示系統中已經安裝的所有修補程式清單。如果新系統中僅安裝一個修補程式(如本範例)，則清單顯示如下：

```
rpm -qPa
pine-4.44-224
```

如果在日後，您想要知道原始安裝的套件版本，可在RPM 資料庫中找到此資訊。以 `pine` 為例，可使用以下指令顯示此資訊：

```
rpm -q --basedon pine
pine = 4.44-188
```

可在 `rpm` 以及 `rpmbuild` 的 `man` 頁面中找到包括RPM 修補程式功能的相關資訊。

4.3.4 Delta RPM 套件

“Delta RPM” 套件包含舊版與新版RPM 套件之間的差異(即“delta”)。在舊RPM 上套用delta RPM 將產生完整的新RPM，但是您不需要一份舊的RPM，delta RPM 也可以和已安裝的RPM 搭配使用。 `deltarpm` 套件比修補程式RPM 的大小還小，如果您必須透過網際網路傳送更新套件，這便是一項優點。其缺點是使用delta RPM 的更新作業將比一般或修補程式RPM 消耗更多的CPU 週期。若要在YOU 工作階段期間讓YaST 使用delta RPM 套件程式，請在 `/etc/sysconfig/onlineupdate` 中將 `YOU_USE_DELTAS` 設為“yes”。

`prepdeltarpm`、`writedeltarpm` 以及 `applydeltarpm` 二進位檔是屬於 `deltarpm` 套裝軟體的一部份，可協助您建立和套用delta RPM 套件。使用下列指令後，即可建立名稱為 `new.delta.rpm` 的delta RPM (即 `old.rpm` 與 `new.rpm` 所存在之處：

```
prepdeltarpm -s seq -i info old.rpm > old.cpio
prepdeltarpm -f new.rpm > new.cpio
```

```
xdelta delta -0 old.cpio new.cpio delta
```

```
writedeltarpm new.rpm delta info new.delta.rpm
rm old.cpio new.cpio delta
```

使用 `applydeltarpm` 即可重新建構新RPM，如果已經安裝舊套件即可從檔案系統：

```
applydeltarpm new.delta.rpm new.rpm
```

或是如果您不想存取檔案系統，而想從舊RPM產生它，就使用-r 選項：

```
applydeltarpm -r old.rpm new.delta.rpm new.rpm
```

如需技術詳細資料，請參閱file:///usr/share/doc/packages/deltarpm/README。

4.3.5 RPM 查詢

使用-q 選項，rpm 會啟動查詢，可檢查RPM 歸檔(藉由新增選項-p)，也可以查詢安裝套件的RPM 資料庫。可使用數個切換來指定所需的資訊類型。請參閱表格4.8, “最重要的RPM 查詢選項”。

Table 4.8: 最重要的RPM 查詢選項

-i	套件資訊
-l	檔案清單
-f FILE	查詢包含(FILE) 檔案的套件(完整的路徑必須以(FILE) 指定)
-s	含有狀態資訊的檔案清單(隱含-l)
-d	只列出文件檔案(隱含-l)
-c	只列出組態檔案(隱含-l)
--dump	含有完整詳細資訊的檔案清單(與-l、-c 或-d 一起搭配使用)
--provides	列出另一個套件可以使用--requires 要求的套件功能
--requires、-R	套件所需的功能
--scripts	安裝程序檔(預先安裝、後續安裝、解除安裝)

例如，rpm -q -i wget 指令可顯示如範例4.2, “rpm -q -i wget” 中所示的資訊。

Example 4.2: rpm -q -i wget

```

Name       : wget                                Relocations: (not relocatable)
Version    : 1.9.1                              Vendor: SUSE LINUX AG, Nuernberg, Germany
Release    : 50                                  Build Date: Sat 02 Oct 2004 03:49:13 AM CEST
Install date: Mon 11 Oct 2004 10:24:56 AM CEST   Build Host: f53.suse.de
Group      : Productivity/Networking/Web/Utilities Source RPM: wget-1.9.1-50.src.rpm
Size       : 1637514                             License: GPL
Signature  : DSA/SHA1, Sat 02 Oct 2004 03:59:56 AM CEST, Key ID a84edae89c800aca
Packager   : http://www.suse.de/feedback
URL        : http://wget.sunsite.dk/
Summary    : A tool for mirroring FTP and HTTP servers
Description:
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
[...]
```

只有在您指定完整檔案名稱及完整路徑時，選項 `-f` 才會有作用。盡可能提供很多檔案名稱。例如，以下指令

```
rpm -q -f /bin/rpm /usr/bin/wget
```

會產生：

```
rpm-4.10.1-191
wget-1.90.1-50
```

如果只知道檔案名稱的一部分，可使用範例4.3, “搜尋套件的程序檔” 中所示的外圍程序程序檔。執行時，可將部份檔案名稱當作參數傳給程序檔。

Example 4.3: 搜尋套件的程序檔

```
#!/bin/sh
for i in $(rpm -q -a -l | grep $1); do
    echo "\"$i\" is in package:"
    rpm -q -f $i
    echo ""
done
```

指令 `rpm -q --changelog rpm` 會顯示特定套件相關資訊的詳細清單(更新、組態、修改等)。此範例顯示套件 `rpm` 的相關資訊。但是，只會列出RPM 資料庫中的最後五筆變更項目。所有項目(回溯到兩年前)均包含在套件本身。只有當CD 1 裝載於 `/media/cdrom` 時，此查詢才有作用：

```
rpm -qp --changelog /media/cdrom/suse/i586/rpm-4*.rpm
```

藉由安裝的RPM 資料庫協助，可執行驗證檢查。以 `-V`、`-y` 或 `--verify` 啓動檢查。使用此選項，`rpm` 可顯示從安裝開始，套件中所有變更過的檔案。`rpm` 使用八個字元的符號來提供下列變更的提示：

Table 4.9: RPM 驗證選項

S	MD5 檢查總數
S	檔案大小
L	符號連結
T	修改時間
D	主要和次要的設備編號
U	擁有者
G	群組
M	模式(許可權和檔案類型)

如果是組態檔，會印出字母c。例如，若/etc/wgetrc (wget) 有變更：

```
rpm -V wget S.5....T c /etc/wgetrc
```

RPM 資料庫的檔案放在/var/lib/rpm。如果分割區/usr的大小為1 GB，此資料庫佔據將近30 MB，尤其是在完整更新之後。如果資料庫遠大於預期，使用選項--rebuilddb來重建資料庫很有用。在執行之前，請備份舊的資料庫。cron 程序檔cron.daily對資料庫做每日備份(以gzip包裝)，並儲存在/var/adm/backup/rpmdb中。備份的數目是由變數MAX_RPMDB_BACKUPS(預設：5)所控制，此變數位於/etc/sysconfig/backup。單一備份的大小大約是1 GB的/usr備份成3 MB。

4.3.6 安裝與編譯來源套件

SUSE LINUX的所有來源套件均帶有副檔名.src.rpm(來源RPM)。

和其他套件一樣，可使用YaST來安裝來源套件。但是，在套件管理員中，它們不會標示為已安裝([i])。這是因為來源套件沒有輸入RPM資料庫中。在您安裝來源套件時，僅會將原始程式碼新增到系統中。軟體必須自行編譯。只有安裝的作業系統軟體會列在RPM資料庫中。

在/usr/src/packages中必須可以找到rpm和rpmbuild的下列目錄(除非您在如/etc/rpmsrc的檔案中指定自訂設定)：

SOURCES 用於原始來源(.tar.bz2 或 .tar.gz 檔案等) 和配送特定調整(大部份是.diff 或.patch 檔案)

SPECS 用於.spec 檔案，和中繼Makefile 相似，可控制build 程序

BUILD 所有來源在此目錄中解壓縮、修補和編譯

RPMS 儲存完整二進位套件的地方

SRPMS 此處為來源 RPM

當您使用YaST 安裝來源套件時，所有需要的元件將會安裝在/usr/src/packages：來源碼與調整在SOURCES 中，而相關的.spec 在SPECS 中。

┌

請勿試驗系統元件(glibc、rpm、sysvinit 等)，因為這樣會危害系統的操作性。

└

以下範例使用wget.src.rpm 套件。在使用YaST 安裝套件之後，應該有類似下列清單的檔案：

```
/usr/src/packages/SOURCES/nops_doc.diff
/usr/src/packages/SOURCES/toplev_destdir.diff
/usr/src/packages/SOURCES/wget-1.9.1+ipvmisc.patch
/usr/src/packages/SOURCES/wget-1.9.1-brokentime.patch
/usr/src/packages/SOURCES/wget-1.9.1-passive_ftp.diff
/usr/src/packages/SOURCES/wget-LFS-20040909.tar.bz2
/usr/src/packages/SOURCES/wget-wrong_charset.patch
/usr/src/packages/SPECS/wget.spec
```

rpmbuild -b <X> /usr/src/packages/SPECS/wget.spec 可開始編譯。(X) 是代表建立程序各種階段的萬用字元(請參閱--help 的輸出或RPM 文件以取得詳細資訊)。以下僅為簡略的說明：

- bp 準備/usr/src/packages/BUILD 中的來源：解壓縮修補程式。
- bc 執行與-bp 相同動作，但是會額外編譯。
- bi 執行與-bp 相同動作，但是會額外安裝內建軟體。警告：如果套件不支援BuildRoot 功能，您可以覆寫組態檔。

- bb** 執行與**-bi** 相同動作，但是會額外建立二進位套件。如果編譯成功，二進位應該在 `/usr/src/packages/RPMS`。
- ba** 執行與**-bb** 相同動作，但是會額外建立來源RPM。如果編譯成功，二進位應該在 `/usr/src/packages/SRPMS`。
- short-circuit** 略過部分步驟。

現在可使用 `rpm -i` 或最好是使用 `rpm -U` 來安裝所建立的二進位RPM。使用 `rpm` 來安裝會讓它出現在RPM 資料庫中。

4.3.7 以build 編譯RPM 套件

許多套件均有一個危機，就是在build 程序中，會有不想要的檔案新增到執行系統中。若要避免發生此狀況，可使用build，它會建立定義的環境，並在其中建立套件。若要建立此chroot 環境，必須提供build 程序檔的完整套件樹狀結構。此樹狀結構可在硬碟上、透過NFS 或從DVD 取得。對應的位置以build `--rpms` `<目錄>` 指定。和rpm 不同，build 指令會尋找來源目錄中的SPEC 檔。若要以裝載在系統中/media/dvd 之下的DVD 重新建立wget (如上面的範例)，請以root 使用者的身分執行下列指令：

```
cd /usr/src/packages/SOURCES/  
mv ../SPECS/wget.spec .  
build --rpms /media/dvd/suse/ wget.spec
```

之後，便會在 `/var/tmp/build-root` 中建立一個最小的環境。套件就會建立在此環境中。完成時，結果套件位於 `/var/tmp/build-root/usr/src/packages/RPMS` 中。

build 程序檔提供數個額外選項。例如，讓程序檔偏好您自己的RPM、省略內建環境的初始化，或將rpm 指令限制在上述階段之一。可使用build `--help` 以及參閱build man 頁面來存取其他資訊。

4.3.8 RPM 歸檔和RPM 資料庫工具

Midnight Commander (mc) 可顯示RPM 歸檔的內容，並複製部分內容。它將歸檔以虛擬檔案系統呈現，提供Midnight Commander 所有通用功能表選項。使用(F3) 可顯示HEADER。使用游標和(Enter) 可檢視歸檔結構。使用(F5) 可複製歸檔元件。

KDE 提供kpackage 工具做為rpm 的前端。完整功能的套件管理員是以YaST 模組的方式提供(請參閱節2.2.1, "安裝和移除軟體")。

系統修復

SUSE LINUX 除了提供各種系統安裝和組態的YaST 模組之外，還提供修復已安裝系統的功能。本章描述系統修復的各種類型和步驟。SUSE 救援系統可用來存取分割區。有經驗的系統管理員可以使用它來修復損毀的系統。

5.1	自動修復	120
5.2	使用者定義的修復	121
5.3	進階工具	122
5.4	SUSE 救援系統	122

由於不能假設損壞的系統可以自行開機，以及執行中的系統無法輕易修復，所以請開機以修復系統，如同您在執行新的安裝一般。請依章1, 使用YaST 安裝所列步驟前往提供各種安裝選項的對話方塊，然後選擇‘修復已安裝系統’。

使用適當的安裝媒體

爲了讓系統在修復之後能夠正常運作，用來啓動系統的安裝媒體應確實符合執行安裝的系統。

下一個步驟是，選擇系統修復的執行方式。可用的方式有自動修復、自訂修復和進階工具，本章將會說明這些方式。

5.1 自動修復

這種方法最適用於修復因不明原因的受損系統。選擇它就會針對已安裝系統啓動周延的分析，由於涉及大量測試和檢查，因此相當費時。程序的進度會以畫面底部的兩個進度列顯示。上面的進度列顯示目前執行測試的進度。下面的進度列顯示分析程序的整體進度。上方區段的記錄視窗會追蹤目前執行中的測試及其結果。請參閱圖形5.1, “自動修復模式”。每次執行時都會執行下列主測試執行。它們包括一些個別子測試。

所有硬碟的分割區表 對所有偵測到之硬碟的分割區表，進行有效性和連貫性的檢查。

交換分割區 會針對已安裝系統的交換分割區進行偵測、測試，並在適用的情況下提議啓用。爲達更高的系統修復速度就會接受這種提議。

檔案系統 所有偵測到的檔案系統都會進行專屬的檔案系統檢查。

檔案/etc/fstab 中的項目 針對檔案中的項目進行完整性和一致性的檢查。所有有效分割區都會裝載上。

開機載入程式組態 針對已安裝系統(GRUB 或LILO) 的開機載入程式組態檢查其完整性和一致性。會檢查開機和root 設備，並檢查initrd 模組的可用性。

套件資料庫 檢查是否具有最小安裝作業所需的所有套件。還有一項可能的選擇性檢查是分析其基本套件。這很耗時，因爲套件的數量十分龐大。

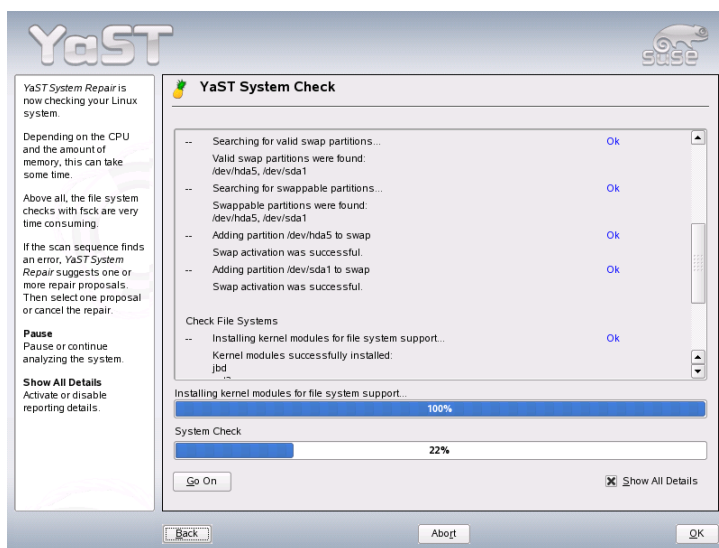


Figure 5.1: 自動修復模式

發生錯誤時，程序停止，並開啓對話方塊，提供詳細資訊及可能的解決方法。各種狀況無法一一詳述。請仔細閱讀畫面上的訊息，並由選項清單上選擇想要的動作。若有疑問，您也可以拒絕提供的修復動作。在此狀況下，系統保持不變，也不會不經提示使用者就自動執行修復。

5.2 使用者定義的修復

前一小節解說的自動修復會執行所有測試。系統損壞程度不明時，這很管用。不過，如果您已經知道系統的哪一部分受到影響，套用的測試範圍可以縮小。選擇‘使用者定義的修復’會顯示測試執行清單，這些測試執行最初都會標記為執行。測試的整個範圍會與自動修復相符。如果您已經知道系統沒有受損，請取消相對應測試的標記。按一下‘繼續’，然後啓動範圍較小的測試程序，它所需的執行時間可能短的多。

不是所有測試群組都可以個別套用。fstab 項目的分析永遠會與檔案系統的檢查同時出現，包括現有的交換分割區。YaST 會選擇所需測試執行的最小數目，以自動滿足這種依存度。

5.3 進階工具

如果您很了解SUSE LINUX，而且已很清楚您的系統需要哪些修復，請選擇‘進階工具’，直接套用修復所需的工具。

安裝新的開機載入程式 這會啟動YaST 開機載入程式組態模組。詳細資訊可以在節8.4, “使用YaST 設定開機載入程式” 中找到。

執行分割工具 這會啟動YaST 裡的進階分割工具。詳細資訊可以在節2.7.5, “磁碟分割” 中找到。

修正檔案系統 這會檢查您已安裝系統中的檔案系統。會先提供所有偵測到之分割區的選單，您就可以選擇一個來進行檢查。

還原遺失的分割區 可以嘗試重建受損的分割區表。首先會提供所偵測到的硬碟清單，供您選擇。按一下‘確定’，啟動檢查。所需時間視處理能力和硬碟大小而定。

重建分割區表

重建分割區表有些麻煩。YaST 會分析硬碟的資料磁區，嘗試辨識遺失的分割區。辨識之後，會將遺失的分割區新增至重新建好的分割區表。不過，這個方法並非對於所有可想像出的例子都會成功。

將系統設定儲存至磁碟 這個選項將重要的系統檔案儲存至軟碟。如果這些檔案中任何一個受損時，就可以由磁片還原。

檢查已安裝的軟體 這會檢查套件資料庫的一致性，和最重要套件的可用性。任何已安裝套件受損時，都可以使用這個工具重新安裝。

5.4 SUSE 救援系統

SUSE LINUX 中有一個救援系統，能在緊急事件時從外部存取Linux 分割區。救援系統可以從CD、網路或SUSE FTP 伺服器載入。救援系統中有多個協助程式，讓您可以解決一些重大問題，例如，無法存取硬碟、設定錯誤的設定檔或其他相似問題。

救援系統的另一個元件是Parted，是用來調整分割區的大小。如果您不想使用整合於YaST 中的調整程式，則可以從救援系統中啓動此程式。關於Parted 的資訊可以在<http://www.gnu.org/software/parted/> 中找到。

5.4.1 啓動救援系統

依照安裝的開機方式將您的系統開機。從開機功能表選取‘救援系統’。接著會解壓縮救援系統，載入RAM 磁碟以做為新的根檔案系統，然後裝載並啓動。

5.4.2 使用救援系統

在(Alt)-(F1) 到(Alt)-(F3) 下，救援系統提供三個虛擬主控台。您可以用root 身份登入而無須密碼。按(Alt)-(F10) 以進入顯示核心與syslog 訊息的系統主控台。

外圍程式與其他許多有用的公用程式，例如mount 程式，都可以在/bin 目錄中找到。sbin 目錄中有重要的檔案與網路公用程式，以供檢視及修復檔案系統之用，包括reiserfsck 與e2fsck。此目錄中也包含最重要的二進位系統維護程式，例如fdisk、mkfs、mkswap、mount、mount、init 及shutdown，還有維護網路的ifconfig、route 以及netstat。/usr/bin 目錄含有vi 編輯器、grep、find、less 以及telnet。

存取一般系統

若要使用救援系統來裝載SUSE LINUX 系統，請使用/mnt 的裝載點。您也可以使用或建立其他目錄。以下範例會以顯示在範例5.1, “/etc/fstab 範例” 中的/etc/fstab 詳細資訊，說明系統的程序。

Example 5.1: /etc/fstab 範例

```
/dev/sdb5    swap    swap    defaults    0    0
/dev/sdb3    /        ext2    defaults    1    1
/dev/sdb6    /usr    ext2    defaults    1    2
```

請注意在下面裝載不同設備一節所概述的步驟順序。

若要存取整個系統，請使用以下的指令將它逐步裝載到 /mnt 目錄中：

```
mount /dev/sdb3 /mnt
mount /dev/sdb6 /mnt/usr
```

現在請存取整個系統，並修正組態檔中的錯誤，例如 /etc/fstab、/etc/passwd 及 /etc/inittab。組態檔現在會位於 /mnt/etc 目錄中，而不是位於 /etc 目錄中。在您藉由重新設定遺失的分割區，並用 fdisk 程式復原該分割區之前，請先列印 /etc/fstab 以及 fdisk -l 的輸出結果。

修復檔案系統

損毀的檔案系統對救援系統來說是難以解決的問題。一般而言，檔案系統無法在執行中的系統上修復。如果發生了嚴重的問題，您可能甚至無法裝載開機檔案，而且系統可能會因為核心異常而無法開機。在此情況下，唯一的方法就是使用救援系統從外部來修復系統。

SUSE LINUX 救援系統中有 reiserfsck、e2fsck 及 dumpe2fs 等公用程式(用來診斷系統)。這些程式應該能解決大部分的問題。緊急狀況之下，通常無法使用 man 頁面。因此，它們包含在此手冊的節 B, "Manual Page of reiserfsck" 與節 B, "Manual Page of e2fsck"。

如果因無效的超級區塊而無法裝載 ext2 檔案系統，則 e2fsck 程式也可能會失敗。如果發生此情況，您的超級區塊也可能會損毀。每 8192 個區塊(8193、16385 等等)都會配置多份超級區塊。如果超級區塊損壞了，請嘗試用其中一個備份來替代。這可藉由輸入 e2fsck -f -b 8193 /dev/damaged_partition 指令來完成。-f 選項會強制檔案系統檢查並覆寫 e2fsck 的錯誤，因為超級區塊的複本仍舊正常，所以運作狀況不會有問題。

Part II

系 統

64 位元系統環境的 32 位元和 64 位元應用程式

SUSE LINUX 可用於多種 64 位元平台。這不表示所有包括的應用程式都已經移植至 64 位元平台。SUSE LINUX 支援在 64 位元系統環境中使用 32 位元應用程式。本章簡略說明這項支援在 64 位元 SUSE LINUX 平台的執行方式。它說明 32 位元應用程式的執行方式(執行期間支援)以及如何編譯 32 位元應用程式，讓它們在 32 位元和 64 位元兩種系統環境都可執行。另外，您可找到關於核心 API 以及 32 位元應用程式如何在 64 位元核心下執行的相關資訊。

6.1	執行期間支援	128
6.2	軟體開發	128
6.3	Biarch 平台的軟體編譯	129
6.4	核心規格	130

SUSE LINUX 針對64 位元平台設計的以及AMD64 和EM64T，讓現有的32 位元應用程式可以在此64 位元環境中執行，且表現出色。這項支援意謂您可以繼續使用偏好的32 位元應用程式，無需等到對應的64 位元連接埠上市。

6.1 執行期間支援

不同應用程式版本之間的衝突

如果應用程式有32 位元和64 位元兩種版本，同時安裝二種版本，一定會發生問題。在這種狀況下，可在兩種版本中選定一種來安裝並使用。

要正確執行，每一個應用程式都需要一些程式庫。不幸的是，這些程式庫的32 位元和64 位元版本，名稱都一樣。必須有其他方法來區分彼此。

要保留與32 位元版本的相容性，程式庫儲存在系統中的位置，與在32 位元環境中的位置相同。在32 位元和64 位元環境中，`libc.so.6` 的32 位元版本都位於 `/lib/libc.so.6`。

所有64 位元程式庫和物件檔案都位於名為 `lib64` 的目錄。您通常預期在 `/lib`、`/usr/lib` 和 `/usr/X11R6/lib` 之下找到的64 位元物件檔案，現在放在 `/lib64`、`/usr/lib64` 以及 `/usr/X11R6/lib64` 底下。這表示在 `/lib`、`/usr/lib` 和 `/usr/X11R6/lib` 之下，有預留空間給32 位元程式庫使用，因而兩種版本的檔案名稱能夠保持不變。

物件目錄中的子目錄，如果其資料內容不取決於文字大小，也不會移動。例如，X11 字型仍然可以在 `/usr/X11R6/lib/X11/fonts` 底下的一般位置找到。此配置與LSB (Linux Standards Base) 以及FHS (File System Hierarchy Standard) 相符。

6.2 軟體開發

Biarich 開發工具鏈可以產生32 位元和64 位元物件。預設是編譯64 位元物件。使用特殊旗標可以產生32 位元物件。GCC 的特殊旗標是 `-m32`。

所有標題檔案必須使用與結構無關的形式來編寫。安裝的32 位元和64 位元程式庫，必須有一個與已安裝標題檔案相符的API (應用程式設計介面)。標準SUSE 環境是根據此原則所設計。對於手動更新的程式庫，請自行解決這些問題。

6.3 Biarch 平台的軟體編譯

若要在biarch架構上，為其他結構開發二進位程式，必須為第二個結構額外安裝相關程式庫。如果第二個結構是32位元結構，則這些套件稱為rpmname-32bit；。您也需要rpmname-devel套件的個別標題和程式庫，以及rpmname-devel-32bit的第二個架構的開發程式庫。

大部份的開放原始碼程式使用的程式組態是以autoconf為基礎。若要使用autoconf為第二個結構設定程式，請覆寫autoconf的一般編譯器和連結器設定，方法是執行帶有其他環境變數的configure程序檔。

以下範例，將x86的AMD64或EM64T系統指示為第二個架構：

1. 將autoconf設定成使用32位元編譯器：

```
CC="gcc -m32"
```

2. 指示連結器處理32位元物件：

```
LD="ld -m elf64_i386"
```

3. 設定組合器來產生32位元物件：

```
AS="gcc -c -m32"
```

4. 判斷libtool等等的程式庫是否來自/usr/lib：

```
LDLFLAGS="-L/usr/lib"
```

5. 判斷程式庫是否儲存在lib子目錄中：

```
--libdir=/usr/lib
```

6. 判斷是否使用32位元X程式庫：

```
--x-libraries=/usr/X11R6/lib/
```

並非每一個程式都需要所有這些變數。將它們配合各程式使用。

```
CC="gcc -m64" \
LDLFLAGS="-L/usr/lib64;" \
    .configure \
    --prefix=/usr \
    --libdir=/usr/lib64
```

```
make
```

```
make install
```

6.4 核心規格

AMD64 以及EM64T 適用的64 位元核心，提供64 位元和32 位元兩種核心ABI (應用程式二進位介面)。後者與相對應32 位元核心的ABI 是相同的。這表示32 位元應用程式可以和64 位元核心溝通，方法與和32 位元核心溝通相同。

64 位元核心中，系統呼叫的32 位元模擬，不支援系統程式使用的某些API。這要視平台而定。因此，有少數應用程式，像`lspci` 或LVM 管理程式，必須如同64 位元程式編譯，才能正常運作。

64 位元核心只可以載入為此核心特別編譯的64 位元核心模組。它無法使用32 位元核心模組。

部份應用程式需要個別的核心可載入式模組。如果您想在64 位元系統環境使用這種32 位元應用程式，請洽詢此應用程式的提供者以及SUSE，確定是否可以取得此模組的核心可載入式模組的64 位元版本以及核心API 的32 位元編譯版本。

啓動及設定Linux系統

Linux 系統的啓動程序複雜。過程中，牽涉到許多不同的元件，和元件之間萬無一失地互動。本章節將提供基礎原則和相關元件的基本說明。本章節也將說明Runlevel 的觀念以及使用sysconfig 進行SUSE 系統組態。

7.1	Linux 開機程序	132
7.2	init 程式	134
7.3	Runlevel	135
7.4	變更Runlevel	136
7.5	Init 程序檔	137
7.6	Systems Services (RunLevel)	140
7.7	SuSEconfig 和 /etc/sysconfig	141
7.8	YaST sysconfig 編輯器	143

7.1 Linux 開機程序

Linux 開機程序由數個階段所組成，每個階段分別由元件所代表。以下清單簡短概述開機程序，以及所有相關主要元件的功能。

1. BIOS

啓動電腦之後，BIOS 會啓始螢幕和鍵盤並測試主記憶體。在此階段中，機器不會存取大量儲存媒體。。接下來，目前日期、時間，以及最重要的週邊設備，將從CMOS 值(CMOS 設定) 載入。識別第一個硬碟及其幾何之後，系統控制將會從BIOS 轉到開機載入器。

2. 開機載入器

第一顆硬碟的第一個實體512 位元組資料磁區，會載入主要記憶體。接著，在此磁區開頭的開機載入器會接管開機程序。開機載入器執行的指令，決定其他部份的開機程序。因此，第一個硬碟的前512 位元組是主開機紀錄(MBR)。然後，開機載入器會將控制傳送到實際作業系統，在這裡是指Linux 核心。Linux 開機載入器的GRUB 詳細資訊，請參閱章8, 開機載入器。

3. 核心和initrd

爲了送出系統控制，開機載入器將核心和初始RAM 磁碟機(initrd) 兩者都載入記憶體。Linux 核心包含具備載入RAM 磁碟機的小型檔案系統的選項，以及裝載實際根目錄檔案系統之前所執行的程式。接著，核心將解壓縮initrd，然後將其裝載爲暫時根目錄檔案系統。Initrd 的內容是一個小型的Linux 系統，其中包含稱爲linuxrc 的可執行檔。在實際根目錄檔案系統裝載之前，會先執行此執行檔。如果可以，核心會釋放initrd 佔用的記憶體，並在成功終止linuxrc 後啓動init。initrd 詳細資訊，請參閱節7.1.1, "initrd"。

4. linuxrc

本程式將執行裝載適當的根目錄檔案系統所需的全部動作，例如提供需要的檔案系統核心功能，並且提供大量儲存控制器設備驅動程式。實際根目錄檔案系統成功裝載後，會立即停止linuxrc，且核心啓動init 程式。如需linuxrc 的詳細資訊，請參閱節7.1.2, "linuxrc"。

5. init

init 透過提供的不同功能來負責系統實際開機的不同層級，init 相關資訊請參閱節7.2, "init 程式"。

7.1.1 initrd

Initrd 是一個小型(通常是壓縮過的)檔案系統，核心可以將其載入RAM 磁碟，並裝載為暫時根目錄檔案系統。在實際根目錄檔案系統裝載之前，它提供可讓程式執行的最小Linux 環境。最小Linux 環境是由BIOS 常式載入記憶體，除了足夠的記憶體之外，沒有特定硬體需求。Initrd 必須隨時提供一個名為linuxrc 的可執行程式，而該程式結束時必須沒有錯誤。

在實際根目錄檔案系統能夠裝載以及實際作業系統可以啟動之前，核心需要相應的驅動程式來存取根目錄檔案系統所在的設備。這些驅動程式可能包含特定類型硬碟的特殊驅動程式，或者甚至包含存取網路檔案系統的網路驅動程式(請參閱)。同時核心也必須包含讀取initrd 檔案系統所需的程式碼。根目錄檔案系統所需的模組可以由linuxrc 載入。

使用程序檔mkinitrd 建立initrd。在SUSE LINUX 中，是透過/etc/sysconfig/kernel 中的INITRD_MODULES 變數來指定要載入的模組。安裝後，此變數會自動設成正確的值(安裝linuxrc 會儲存載入哪些模組)。模組會確實依據它們在INITRD_MODULES 中出現的順序來載入。如果使用到多個SCSI 驅動程式時，這點尤其重要，因為不這麼做，硬碟名稱會有所變更。嚴格來說，載入那些存取根目錄檔案系統所需的驅動程式便足夠了。然而，因為後續載入可能會有問題，所以所有安裝所需的SCSI 驅動程式都將透過initrd 載入。

更新initrd

開機載入器將採用與核心相同的方式載入initrd。更新initrd 之後不需要重新安裝GRUB，因為開機時GRUB 會在目錄搜尋正確的檔案。

7.1.2 linuxrc

Linuxrc 的主要目的是準備裝載實際根目錄檔案系統，以及存取實際根目錄檔案系統。根據您實際系統組態，linuxrc 負責下列工作。

載入核心模組 根據硬體組態，存取您電腦的硬體元件可能需要特殊的驅動程式(最重要的元件是硬碟)。若要存取最後根目錄檔案系統，核心需載入適當的檔案系統驅動程式。

管理RAID 和LVM 設定 如果您將系統設定為RAID 或LVM 之下的根目錄檔案系統，linuxrc 會設定LVM 或RAID，以便之後能夠存取檔案根目錄系統。RAID 相關資訊請參閱節3.8, "軟體RAID 組態"。LVM 相關資訊請參閱節3.7, "LVM 組態"。

管理網路組態 如果您將系統設為使用網路裝載的根目錄檔案系統(透過NFS裝載)，則linuxrc 必須確認是否載入了適當的網路驅動程式，還有是否設定為允許存取根目錄檔案系統。

在初始開機時呼叫linuxrc 為安裝過程的一部份，該工作不同於先前所述。

尋找安裝媒體 啓動安裝程序時，您的機器會從安裝媒體使用YaST 安裝程式，來載入一個安裝核心，以及一個特殊initrd。在RAM 檔案系統中執行的YaST 安裝程式，必需具備安裝媒體的實際位置相關資訊，以便存取該程式和安裝作業系統。

啓始硬體辨識並載入適當核心模組 如同在節7.1.1, “initrd” 所述，開機程序會以最少驅動程式啓動，供大部份的硬體組態使用。linuxrc 將啓動一個硬體掃描程序，該程序會判斷驅動程式是否適用您的硬體組態。這些值稍後會寫入/etc/sysconfig/kernel 檔案中的INITRD_MODULES，讓後續開機程序能夠使用自訂initrd。安裝程序進行期間，linuxrc 會載入此模組。

載入安裝系統或救援系統 在硬體妥善識別及載入適當驅動程式後，linuxrc 會啓動安裝系統，其中包含實際YaST 安裝程式或救援系統。

啓動YaST 最後，linuxrc 將會啓動YaST，而YaST 會啓動套件安裝和系統組態。

7.1.3 若需詳細資訊

如需詳細資訊，請參閱/usr/src/linux/Documentation/ramdisk.txt、/usr/src/linux/Documentation/initrd.txt，以及initrd(4)和mkinitrd(8)的man 頁面。

7.2 init 程式

init 程式的程序編號是1，負責以要求的方式啓始系統。所有其他程序是init 的子程序，或它其中一個子代。init 的角色很特殊。它由核心直接啓動並拒絕訊息9，這個訊息通常會刪除程序。所有其他程式是直接透過init 或它其中一個子程序啓動。

init 是在/etc/inittab 檔案中設定組態。runlevels 就定義於此處(請參閱節7.3, “Runlevel”)。它也指定每一個層次可以使用的服務和精靈。視/etc/inittab 中的項目而定，init 會執行數個程序檔。為避免混淆，這些程序檔都位於目錄/etc/init.d 中。

啟動系統和關閉系統的整個程序是由init維護。依此觀點，核心可以視為背景程序，它的工作是維護所有其他程序，並根據其他程式的要求，調整CPU時間和硬體存取。

7.3 Runlevel

在Linux中是由runlevel定義啟動系統的方式，以及在所執行的系統可以使用哪些服務。開機之後，系統會按照/etc/inittab中的initdefault這一行文字的定義而啟動。一般是3或5。請參閱表格7.1, “可用的Runlevel”。還有一種方法是，runlevel可以在開機期間指定(例如，在開機提示時)。非經由核心直接評估的所有參數，都會傳送給init。

若要在系統執行時變更runlevel，請輸入init以及當成引數的對應數字。只有系統管理員可執行此動作。init 1(或shutdown now)會導致系統變更為單一使用模式，該模式用於系統維護和系統管理。管理員完成工作之後，可以輸入init 3，回到正常的runlevel，它會啟動所有基礎程式並允許一般使用者登入和使用系統，沒有X環境。若要啓用圖形環境，例如GNOME、KDE或任何其他視窗管理員，請使用init 5。而init 0(或shutdown -h now)會使系統停止。init 6(或shutdown -r now)會關閉系統並重新啓動。

透過NFS裝載/usr分割區的Runlevel 2

如果系統透過NFS掛載/usr分割區，則不應該使用runlevel 2。/usr目錄保存有系統正常作業所不可或缺的重要程式。因為runlevel 2(本地多重使用者模式，無遠端網路)無法使用NFS服務，所以系統在很多方面都會受到嚴格限制。

Table 7.1: 可用的Runlevel

Runlevel	說明
0	系統暫停
S	單一使用者模式；從開機提示，只能使用使用美式鍵盤對應
1	單一使用者模式
2	本地多重使用者，不包含遠端網路(例如NFS)
3	完整的多重使用者模式，包含網路

4	未使用
5	完整多重使用者模式，包含網路、X 顯示管理員—KDM (預設)、GDM 或XDM
6	系統重新開機

所有SUSE LINUX 標準安裝中，預設都使用runlevel 5。會提示使用者直接在圖形介面下登入。如果預設runlevel是3，必須如章11, X Window System 所述，適當設定X視窗系統，runlevel才可以切換至5。這個動作完成之後，請輸入init 5，檢查系統是否會依照要求的方式運作。如果一切都如預期，您可以使用YaST，將預設runlevel設定成5。

└

修改/etc/inittab

如果/etc/inittab損毀，系統可能無法正常開機。因此，編輯/etc/inittab時要十分小心，並且永遠要備份完整的版本。若要修復損毀，試著在開機提示的核心名稱後面輸入init=/bin/sh，直接開機至外圍程序。接著，使用指令mount -o remount,rw / 讓您的根目錄檔案系統可寫入，然後，使用cp指令，以備份的版本取代/etc/inittab。若要預防檔案系統發生錯誤，請在重新開機之前，將根目錄檔案系統變更為唯讀：mount -o remount,ro /。

└

7.4 變更Runlevel

通常，當您變更runlevel時會發生兩件事。首先，啟動目前runlevel中的停止程序檔，關閉對目前runlevel很重要的一些程式。然後啟動新runlevel的啟動程序檔。在大部份的情況下，此時也會啟動一些程式。例如，從runlevel 3變更成5時，會發生以下事件：

1. 管理員(root)可以輸入init 5，要求init變更為不同的runlevel。
2. init會洽詢它的組態檔(/etc/inittab)然後決定是否以新的runlevel為參數來啟動/etc/init.d/rc。
3. 現在rc會呼叫目前runlevel的所有停止程序檔，但只限新的runlevel沒有啟動的程序檔。在此範例中的所有程序檔，都位於/etc/init.d/rc3.d

(舊的runlevel是3)，而且開頭是K。K後面的號碼指定啓動順序，因為還有一些相關因素要考量。

4. 新的runlevel啓動程序檔，會最後才啓動。在此範例中的所有程序檔，都位於/etc/init.d/rc5.d，而且開頭是S。啓動順序的相同程序，也會適用於此。

變更成與目前runlevel相同的runlevel時，init只會檢查/etc/inittab是否變更，並啓動適當的步驟，例如，在另一個介面啓動getty。

7.5 Init 程序檔

/etc/init.d 中的程序檔有兩種類型：

由init直接執行的程序檔 這種狀況僅出現於開機程序或立即關閉系統(電源中斷或使用者按下`Ctrl-Alt-Del`)時。此程序檔定義於/etc/inittab。

由init間接執行的程序檔 這些程序檔在變更runlevel時就會執行，而且永遠會呼叫主要程序檔/etc/init.d/rc，以保證相關程序檔的順序正確。

所有程序檔都位於/etc/init.d。也可在這裡找到變更runlevel的程序檔，不過要從其中一個子目錄(/etc/init.d/rc0.d到/etc/init.d/rc6.d)，透過符號連結來呼叫。這樣的安排是爲了明確，以避免程序檔用於多個runlevel時，造成重複情況。因爲每一個程序檔都可當成啓動程序檔和停止程序檔來執行，所以這些程序檔必須了解參數start和stop。程序檔也了解restart、reload、force-reload和status選項。這些不同選項在表格7.2，“可能的init程序檔選項”都有說明。直接由init執行的程序檔沒有這些連結。需要時，可從runlevel獨立執行這些選項。

Table 7.2: 可能的init程序檔選項

選項	說明
start	啓動服務。不做任何動作也可以成功啓動執行的服務。
stop	停止服務。
restart	如果服務在執行中，先停止，再重新啓動。如果服務沒有執行，請啓動它。

<code>reload</code>	不需停止和重新啓動服務，就可以重新載入組態。
<code>force-reload</code>	如果服務支援，請重新載入組態。否則，執行與 <code>restart</code> 相同的動作。
<code>status</code>	顯示目前狀態。

每一個特定runlevel子目錄中的連結，可以將程序檔與不同runlevel產生關聯。安裝或解除安裝套件時，可以透過程式`insserv`的協助，新增和移除這些連結(或透過`/usr/lib/lsb/install_initd`，它是一個會呼叫該程式的程序檔)。若需詳細資訊，請參閱man頁面的`insserv(8)`。簡介最先啓動的開機及最後啓動的停止程序檔，並說明維護程序檔。

使用`init`直接啓動系統時會執行。它與所選的runlevel無關，而且只會執行一次。此時，會裝載`proc`和`pts`檔案系統，啓動`blogd`(開機記錄精靈)。如果系統是更新或安裝之後第一次開機，會啓動啓始系統組態。

任何其他服務啓動之前，`blogd`精靈是透過開機和`rc`啓動的服務。上述程序檔觸發的動作(例如，執行一些子程序檔)完成後，`blogd`精靈就會停止，並將螢幕的所有輸出寫入`/var/log/boot.msg`記錄檔，但這只發生在`/var`裝載成讀寫時。否則，`blogd`會緩衝處理所有螢幕資料，直到`/var`可以使用為止。在man頁面的`blogd(8)`，可取得進一步資訊。

程序檔`boot`也負責啓動`/etc/init.d/boot.d`中，名稱開頭是`S`的所有程序檔。在該處，會檢查檔案系統，並在需要時設定迴圈設備。也會設定系統時間。如果自動檢查和修復檔案系統時發生錯誤，系統管理員輸入管理員密碼後即可介入。最後執行的是程序檔`boot.local`。

`boot.local` 在此輸入開機時要執行的其他指令(進入runlevel之前)。它就像是DOS系統的`AUTOEXEC.BAT`。

`boot.setup` 從單一使用者模式變更成任何其他runlevel時會執行此程序檔，它負責一些基本設定，例如鍵盤配置和虛擬主控台的啓始化。

`halt` 只有進入runlevel 0或6時，才會執行此程序檔。在此，它是以`halt`或`reboot`的形式執行。系統是否關閉或重新啓動，取決於呼叫`halt`的方式。

`rc` 此程序檔會呼叫目前runlevel的適當停止程序檔，以及新選取runlevel的啓動程序檔。

7.5.1 新增init 程序檔

您可以建立自己的程序檔，並輕鬆將它們整合至上述配置。如需關於格式化、命名以及組織自訂程序檔的說明，請參閱LSB的規格和man頁面的init(8)、init.d nit.d(7)以及insserv(8)。此外，請參閱man頁面的startproc(8)和killproc(8)。

┌

建立自己的init 程序檔

而言，我不知道什麼樣的說法適當。錯誤init程序檔可能會停止您的機器。編輯類似程序檔要格外小心，可能的話，讓它們在多重使用者環境下密集測試。關於init程序檔的一些有用資訊，請參閱節7.3, “Runlevel”。

└

若要為指定的程式或服務建立自訂init程序檔，請將檔案/etc/init.d/skeleton當成樣板。使用新名稱儲存此檔案副本，編輯所需相關程式和檔案名稱、路徑，以及其他詳細資料。您也可能需要使用自己的組件來強化程式檔，好讓init程序觸發正確的動作。

上方的INIT INFO區塊是程序檔的必要組件，應該要編輯。請參閱範例7.1, “迷你INIT INFO 區塊”。

Example 7.1: 迷你INIT INFO 區塊

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

請在INFO區塊的第一行中、Provides:後面，為此init程序檔所控制的程式或服務指定名稱。在Required-Start:和Required-Stop:這兩行，在服務本身啟動或停止前，指定要啟動或停止的所有服務。此資訊以後會用來產生程序檔名稱的編號，這些就是在runlevel目錄可以找到的編號。在Default-Start:和Default-Stop:下方，指定應該自動啟動或停止服務的runlevel。最後，在Description:下方，簡要說明討論中的服務。

若要建立由runlevel 目錄(/etc/init.d/rc?.d/) 對應到 /etc/init.d/ 相關程序檔的連結，請輸入指令 `insserv <new-script-name>`。insserv 程式會評估 INIT INFO 標題，為runlevel 目錄(/etc/init.d/rc?.d/) 中的啟動和停止程序檔建立必要的連結。該程式也會為這些連結的名稱加上必要的編號，即可依正確順序啟動和停止每一個runlevel。如果您偏好以圖形工具來建立類似連結，請使用YaST 提供的runlevel 編輯器，請見節7.6, "Systems Services (RunLevel)"。

如果/etc/init.d/ 裡已經有程序檔，應該整合至現有的runlevel 配置，使用insserv 在runlevel 目錄中立即建立連結，或者在YaST 的runlevel 編輯器中啓用對應的服務。下次重新啟動時，會套用您所做的變更—新服務會自動啟動。不要手動設定這些連結。如果INFO 區塊發生錯誤，稍後執行某些其他服務的insserv 將會發生問題。

7.6 Systems Services (RunLevel)

啟動YaST 模組後，它會顯示概觀，列示所有可用的服務，以及每個服務目前的狀態—無論它們是否已啓用。決定要以‘簡單模式’或‘進階模式’使用模組。大部份情況下，預設的‘簡單模式’應該都已夠用。左欄顯示服務的名稱，中間欄顯示它的目前狀態，而右欄提供簡短說明。視窗下方為選取的服務提供更詳細的說明。若要啓用服務，在表格中選取它，然後選取‘啓用’。停用服務的步驟也一樣。

要更仔細控制啟動或停止服務的runlevel，或者變更預設的runlevel，請先選取‘進階模式’。在此模式下，畫面上方的對話方塊會顯示目前的預設runlevel 或“initdefault” (系統預設啟動的runlevel)。一般情況下，SUSE LINUX 系統的預設runlevel 是runlevel 5 (含網路和X 的完整多重使用者模式)。合適的替代方法可能是runlevel 3 (含網路的完整多重使用者模式)。

此YaST 對話方塊允許其中一個runlevel 選項(如表格7.1, "可用的Runlevel" 所列) 做為新預設。還可使用此視窗中的表格，啓用或停用個別服務和精靈。表格會列示可用的服務和精靈，顯示目前在您的系統上它們是否啓用，如果啓用，是哪一個runlevel。使用滑鼠選取其中一列後，按一下代表runlevel 的核取方塊('B'、'0'、'1'、'2'、'3'、'5'、'6' 和'S') 來定義runlevel，以便在該處執行選取的服務或精靈。Runlevel 4 一開始並未定義，以便建立自訂runlevel。表格概觀的正下方，提供目前所選服務或精靈的簡要說明。

使用‘啓動、停止或重新整理’，決定是否啟動服務。‘重新整理狀態’ 檢查目前狀態。您可以使用‘設定或重設’ 來做選擇，將您的變更套用至系統，或復原啟動runlevel 編輯器之前的設定。選取‘完成’，就會將變更的設定儲存至磁碟。

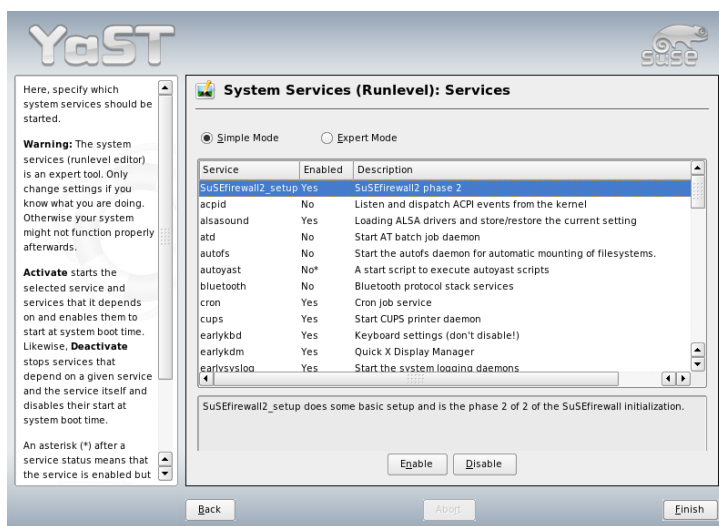


Figure 7.1: Systems Services (RunLevel)

┌

變更Runlevel 設定

錯誤的runlevel 設定可能會造成系統無法使用。在您套用變更之前，請務必確定您知道它們的後果。

└

7.7 SuSEconfig 和/etc/sysconfig

SUSE LINUX 的主要組態可以用/etc/sysconfig 的組態檔來建立。/etc/sysconfig 中個別的檔案只由相關的程式檔讀取。這可以確保例如網路設定，只由網路相關的程序檔來剖析。很多其他系統組態檔是根據/etc/sysconfig 的設定而產生的。此工作是由SuSEconfig 執行。例如，如果您變更網路組態，SuSEconfig 就可能也會變更檔案/etc/host.conf，因為這是與網路組態相關的一個檔案。

如果您手動變更這些檔案中的任何資料，請於之後執行SuSEconfig，確定所有相關位置的必要變更都已完成。如果您使用YaST sysconfig 編輯器變更組態，會自動套用所有變更，因為YaST 會在需要時自動啓動SuSEconfig，更新組態檔。

這種概念讓您可以對組態進行基本變更，不必重新啓動系統。因為部份變更相當複雜，所以某些程式必須重新啓動，變更才會生效。例如，網路組態的變更可能需要重新啓動相關網路程式。輸入指令rcnetwork stop 和rcnetwork start，可以讓變更的網路組態生效。

建議變更系統組態的方法，包含以下步驟：

1. 使用init 1，將系統置於單一使用者模式(runlevel 1)。
2. 視需要變更組態檔。使用選擇的編輯器或YaST sysconfig 編輯器，就可以達到目的(請參閱節7.8, "YaST sysconfig 編輯器")。

┌

手動變更系統組態

如果未使用YaST 來變更/etc/sysconfig 中的組態檔，請確定空的變數值是由兩個引號所表示(KEYTABLE="")，而且該值與其中的空格包夾在引號中。由一個文字組成的值不需要包夾在引號中。

└

3. 執行SuSEconfig 來確定變更生效。如果使用YaST 變更組態檔，會自動完成這個步驟。
4. 使用像init 3 (取代3 為之前的runlevel) 的指令，將您的系統置回之前的runlevel。

變更整個系統設定，例如網路組態時，才會用到此程序。小幅度變更不必進入單一使用者模式，不過您也可以進入單一使用者模式，即可百分之百確定所有相關程式都正確重新啓動。

設定自動系統組態

若要使用SuSEconfig 關閉自動系統組態，將/etc/sysconfig/suseconfig 的ENABLE_SUSECONFIG 設為no。如果您想使用SUSE 安裝支援，請勿停用SuSEconfig也可以停用部份自動組態。

7.8 YaST sysconfig 編輯器

儲存最重要的SUSE LINUX 設定之檔案位於 `/etc/sysconfig` 目錄。sysconfig 編輯器會以易讀的方式顯示選項。值可以修改，然後新增至此目錄中的個別組態檔。不過，一般而言，不需要手動編輯它們，因為安裝套件或設定服務時，就會自動調整這些檔案。

┌

修改 `/etc/sysconfig/*` 檔案

如果您先前並無經驗和相關知識，請勿修改 `/etc/sysconfig` 檔案。它可能會嚴重破壞您的系統。`/etc/sysconfig` 中的檔案包含每一個變數的簡短註解，說明它們的實際作用。

└

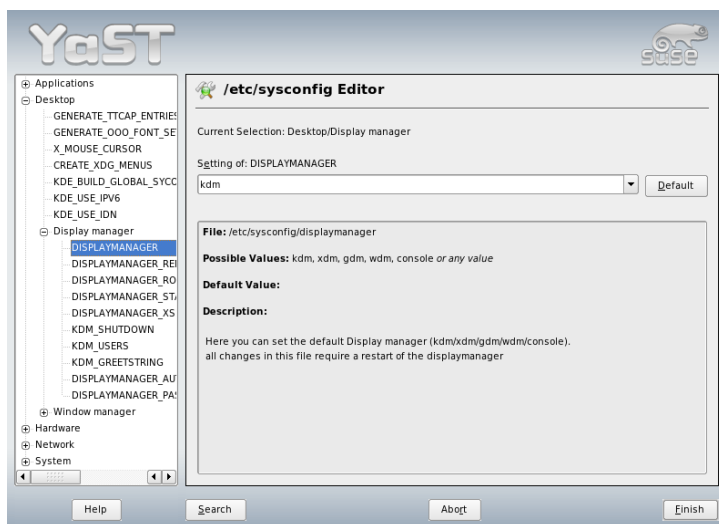


Figure 7.2: 使用sysconfig 編輯器的系統組態

YaST sysconfig 對話方塊分割成三個部份。對話方塊的左側顯示所有可設定變數的樹狀結構檢視。當您選取變數時，右側會顯示目前的選擇，以及此變數的目前設定。在下方，第三個視窗顯示變數用途的簡短說明、可能值、預設值以及產生

此變數的實際組態檔。對話方塊也提供關於變更變數後要執行的組態程序檔，以及變更結果會啓動什麼新服務等等的相關資訊。YaST 會要求您確認變更，並告訴您在選取‘完成’並離開對話方塊後會執行的程序檔。請選取目前要略過的服務和程序檔，它們會在稍後啓動。

開機載入器

本章說明如何設定GRUB，這是SUSE LINUX使用的開機載入器。一種特殊的YaST模組可以用於執行所有設定。如果您不熟悉Linux開機的要點，請閱讀以下章節，取得部份背景資訊。本章節也敘述使用GRUB開機時，經常遇到的部份問題以及它們的解決方案。

8.1	開機管理	146
8.2	選取開機載入器	147
8.3	使用GRUB開機	147
8.4	使用YaST設定開機載入程式	155
8.5	解除安裝Linux開機載入器	158
8.6	建立開機CD	158
8.7	圖形化SUSE畫面	159
8.8	疑難排解	160
8.9	若需詳細資訊	161

本章節著重於開機管理以及開機載入器GRUB的組態。整個開機程序簡述於章7, 啓動及設定Linux系統。開機載入器是機器(BIOS)與作業系統之間的介面(SUSE LINUX)。開機載入器的組態會決定要啓動的作業系統及其選項。

下列詞彙將在本章節中經常出現，而且可能需要說明：

主開機記錄 MBR的結構是由作業系統(獨立會議)所定義。前446位元組是保留給程式碼。它們通常具有開機載入器程式，在這個例子中為GRUB。接下來的64位元組提供多達四個分割區表格的空間(請參閱節1.5.4, “分割區類型”)。分割區表格包含硬碟的分割以及檔案系統類型相關資訊。作業系統需要此表格來處理硬碟。MBR的最後兩個位元組必須包含靜態的“魔術編號”(AA55)。包含不同值的MBR, BIOS和所有個人電腦作業系統都會被視為無效。

開機磁區 開機磁區是除了擴充分割區以外，硬碟分割區的第一個磁區，它只會當成其他分割區的“容器”。這些開機磁區有512位元組的空間供程式碼使用，而這些程式碼是用來啓動各自分割區中安裝的作業系統。此適用於DOS、Windows和OS/2分割區格式的開機磁區，它們也包含檔案系統的部份重要基本資料。相反的，Linux分割區的開機磁區一開始是空白，直到設定檔案系統之後，才會寫入記錄。因此，Linux分割區無法自行啓動，即使它包含核心以及有效的根檔案系統。開機磁區如果包含可以啓動系統的有效程式碼，則它的魔術編號與MBR的最後兩個位元組相同(AA55)。

8.1 開機管理

在最簡單的案例中——如果只有一個作業系統安裝在電腦上——開機管理會如上述般進行。如果多個作業系統安裝在電腦上，可以使用以下選項：

從外部媒體啓動其他系統 其中一個作業系統可以從硬碟啓動。其他作業系統是透過安裝在外部媒體的開機管理員啓動的(磁片、USB儲存媒體)。因為GRUB可以啓動所有其他作業系統，所以不需要使用外部開機載入器。

在MBR安裝開機管理員 開機管理員可以在一部電腦上進行同時安裝並交替使用多個系統。使用者可以在開機程序期間選取要啓動的系統。若要變更至其他系統，必須重新啓動電腦。只有選取的開機管理員與安裝的作業系統相容時，才有可能。GRUB - SUSE LINUX使用的開機管理員，可以啓動所有常見的作業系統。就預設值，SUSE LINUX會在MBR安裝選取的開機管理員。

8.2 選取開機載入器

就預設值，SUSE LINUX 會使用GRUB開機載入器。不過，有時候以及在特殊軟、硬體的配合，LILO可能更合適。如果您更新使用LILO的舊版SUSE LINUX，則安裝LILO。至於新的安裝，會安裝GRUB，除非根分割區安裝在以下系統：

- CPU相關的RAID控制器(例如很多Promise或Highpoint控制器)
- 軟體RAID
- LVM

如需關於安裝和設定LILO的資訊，請參閱「支援資料庫」中的關鍵字LILO。

8.3 使用GRUB開機

GRUB (Grand Unified Bootloader) 包含兩個階段。第一個階段包含512個位元組並寫入MBR或硬碟分割區或磁片的開機磁區。接下來，會載入第二個階段。此階段包含實際的程式碼。第一個階段的唯一工作是載入開機載入器的第二個階段。

stage2可以存取檔案系統。目前支援，Ext2、Ext3、ReiserFS、Minix和Windows使用的DOS FAT檔案系統。對於JFS、XFS、UFS以及BSD使用的FFS，也支援到一定的程度。自0.95版開始，GRUB也可以從包含符合“El Torito”規格的ISO 9660標準檔案系統的CD或DVD啓動。即使在系統啓動之前，GRUB可以存取支援的BIOS磁碟機(BIOS偵測到的磁片或硬碟、CD光碟機和DVD磁碟機)的系統。因此，對GRUB組態檔(menu.lst)所做的變更，將不再需要重新安裝開機管理員。啓動系統後，GRUB會重新載入功能表檔案以及核心或起始RAM磁碟(initrd)的有效路徑和分割區資料，然後找這些檔案。

GRUB的實際組態是以下列敘述的三個檔案為基礎：

`/boot/grub/menu.lst` 此檔案包含可以使用GRUB啓動的分割區或作業系統，所有的相關資訊。沒有此資訊，系統控制權便無法傳送給作業系統。

`/boot/grub/device.map` 此檔案會從GRUB和BIOS表示法，將設備名稱轉譯成Linux設備名稱。

`/etc/grub.conf` 此檔案包含GRUB外圍程序正確安裝開機載入器時需要的參數和選項。

有多種方法可以控制GRUB。現有組態啓動項目，可以從圖形功能表選取(開頭顯示畫面)。組態會從檔案menu.lst 載入。

在GRUB，啓動前可以變更所有啓動參數。例如，編輯功能表檔案發生的錯誤，可以用此方法更正。啓動指令也可以透過各種輸入提示，以互動方式輸入(請參閱節8.3.1, “在開機程序期間編輯功能表項目”)。GRUB 提供啓動前，判斷核心和initrd 二者位置的可能性。以此方法，您還可以爲開機載入器組態中不存在的項目，啓動安裝的作業系統。

GRUB 外圍程序在安裝的系統中，提供GRUB 的模組。它可以在實際套用之前，安裝GRUB 或測試新設定。請參閱節8.3.4, “GRUB 外圍程序”。

8.3.1 GRUB 開機功能表

圖形開頭顯示畫面以及開機功能表是以GRUB 組態檔/boot/grub/menu.lst 爲基礎的，它包含可以透過功能表啓動的所有分割區或作業系統，全部的相關資訊。

每次啓動系統時，GRUB 會從檔案系統載入功能表檔案。基於此因素，GRUB 不需要在每次變更檔案後，重新安裝。使用YaST 開機載入器來修改GRUB 組態，如節8.4, “使用YaST 設定開機載入程式” 所述一般。

功能表檔案包含指令。語法相當簡單。每一行包含一個指令，後面是由像外圍程序中的空格所分開的選用參數。基於歷史因素，部份指令可以允許= 放在第一個參數前面。註解是以井字號(#) 開頭。

要識別功能表概觀中的功能表項目，爲每一個項目指定標題。關鍵字標題後面的文字(包括任何空格) 會在功能表中顯示成可選取的選項。下一個標題的所有指令，會在選取此功能表項目時執行。

最簡單的案例是重新導向至其他作業系統的開機載入器。指令是chainloader 及引數通常是GRUB 區塊表示法中，其他分割區的啓動區塊。例如：

```
chainloader (hd0,3)+1
```

GRUB 中的設備名稱會在節8.3.1, “硬碟和分割區的命名慣例” 說明。上述範例指定第一個硬碟中，第四個分割區的第一個區塊。

使用指令kernel 來指定核心影像。第一個引數是分割區中，核心影像的路徑。其他引數會傳送到指令行上的核心。

如果核心沒有內建的驅動程式可存取根分割區，initrd 必須以個別的GRUB 指令來指定，它唯一的引數是initrd 檔案的路徑。因爲initrd 的載入位址是記錄於載入的核心影像，所以指令initrd 必須立即接在kernel 指令的後面。

指令`root`會簡化核心和`initrd`檔案的指定。`root`的唯一引數是GRUB設備或GRUB設備上的分割區。此設備是用於所有核心、`initrd`或其它檔案路徑，就檔案路徑而言，直到下一個`root`指令下達時，才會明確指定設備。此指令不是用於安裝時產生的`menu.lst`檔案。它只是利於手動編輯。

`boot`指令會在每一個功能表項目最後暗示，所以它不需要寫入功能表檔案。不過，如果您使用互動式GRUB來啟動，必須在最後輸入`boot`指令。指令本身沒有引數。它只是啟動載入的核心影像或指定的鏈結載入器。

撰寫所有功能表項目之後，將其中一個定義為預設項目。否則，會使用第一個(項目0)做為預設項目。您也可以指定預設項目要在幾秒後啟動。`timeout`和`default`通常在功能表項目前面。節8.3.1, “範例功能表檔案”會敘述一個範例檔案。

硬碟和分割區的命名慣例

硬碟和分割區使用的命名慣例GRUB與一般Linux設備使用的命名慣例不同。在GRUB，分割區的編號會從0開始。因此，`(hd0,0)`是第一個硬碟的第一個分割區。與一般桌上型機器的primary master硬碟對應的Linux設備名稱是`/dev/hda1`。

四個可能的主要分割區會指定分割區編號0到3。邏輯分割區是從4開始編號：

<code>(hd0,0)</code>	第一個硬碟的主分割區
<code>(hd0,1)</code>	第二個主分割區
<code>(hd0,2)</code>	第三個主分割區
<code>(hd0,3)</code>	第四個主分割區 (一般是擴充分割區)
<code>(hd0,4)</code>	第一個邏輯分割區
<code>(hd0,5)</code>	第二個邏輯分割區

GRUB不會區分IDE、SCSI和RAID設備。BIOS或其他控制器識別的所有硬碟，會按照BIOS的啟動順序編號。

可惜的是GRUB無法將Linux設備名稱正確對應至BIOS設備名稱。它會透過演算法的協助來產生此對應，然後儲存至檔案`device.map`，需要時可以編輯它。如需關於檔案`device.map`的資訊，請參閱節8.3.2, “檔案`device.map`”。

完整GRUB路徑包含放在括號中的設備名稱，以及指定分割區中檔案系統的檔案路徑。路徑開頭是一個斜線。例如，如果系統有一個IDE硬碟，它的第一個分割區包含Linux，則可以使用下列方式設定核心：

```
(hd0,0)/boot/vmlinuz
```

範例功能表檔案

以下範例顯示GRUB 功能表檔案的結構。範例安裝包含 /dev/hda5 下的Linux 啓動分割區、/dev/hda7 下的根分割區，以及 /dev/hda1 下的Windows 安裝。

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd

title windows
    chainloader(hd0,0)+1

title floppy
    chainloader(fd0)+1

title failsafe
    kernel (hd0,4)/vmlinuz.shipped root=/dev/hda7 ide=nodma \
    apm=off acpi=off vga=normal nosmp maxcpus=0 3
    initrd (hd0,4)/initrd.shipped
```

第一個區塊定義開頭顯示畫面的組態：

gfxmenu (hd0,4)/message 背景影像message 是位在 /dev/hda5 。

color white/blue black/light-gray 色彩配置：白色(前景)、藍色(背景)，黑色(選擇項目)，以及淺灰色(選擇項目的背景)。色彩配置對於開頭顯示畫面沒有作用，只對您可以存取而且可以自訂的GRUB 功能表有作用(您可以使用 **Esc**，結束開頭顯示畫面)。

default 0 第一個功能表項目 **title linux** 是預設要啓動的項目。

timeout 8 沒有任何用者輸入，經過8 秒後，GRUB 會自動啓動預設項目。

第二個(最大的) 區塊會列示各種可啓動的作業系統。個別作業系統的章節是由 **title** 開始。

- 第一個項目(title linux)負責啓動SUSE LINUX。核心(vmlinuz)是位在第一個硬碟的第一個邏輯分割區(啓動分割區)。核心參數，例如根分割區和VGA模式，會在此添加。根分割區是根據Linux命名慣例(/dev/hda7/)指定的，因為此資訊是由核心讀取，與GRUB無關。initrd也是位在第一個硬碟的第一個邏輯分割區。
- 第二個項目負責載入Windows。Windows是從第一個硬碟的第一個分割區啓動(hd0, 0)。指令chainloader +1會造成GRUB載入並執行指定分割區的第一個磁區。
- 下一個項目允許從磁片啓動，無需修改BIOS設定。
- 開機選項failsafe會以選擇的核心參數來啓動Linux，可以啓動發生問題的Linux系統。

功能表檔案可以在需要時變更。GRUB會在下次啓動時使用修改的設定。使用YaST或選擇的編輯器，永久地編輯檔案。另一種方法是，使用GRUB的編輯功能，以互動方式暫時變更。請參閱節8.3.1, “在開機程序期間編輯功能表項目”。

在開機程序期間編輯功能表項目

在圖形化GRUB開機功能表中，可以使用方向鍵選取要啓動的作業系統。如果選取Linux系統，可以在開機提示時輸入其他啓動參數。要完全直接編輯個別功能表，按(Esc)結束開頭顯示畫面，然後按(E)。用此方式所做的變更，只會套用於目前的開機程序，不會永久變更。

開機程序期間的鍵盤配置

US鍵盤配置是啓動時唯一可以使用的鍵盤配置。

啓動編輯模式之後，使用方向鍵選取功能表項目來編輯組態。要讓組態變成可以編輯，再按(E)一次。使用此方法，在對於開機程序產生負面影響之前，編輯錯誤的分割區或路徑指定。按(Enter)，結束編輯模式並返回功能表。然後按(B)來啓動此項目。底下的說明文字會顯示進一步可行的動作。

要永久地輸入變更的開機選項，然後傳送至核心，以root身份開啓檔案menu.lst，然後將各自的核心參數添加至現有的行，以空格分隔：

```
title linux kernel (hd0,0)/vmlinuz root=/dev/hda3 additional parameter
initrd (hd0,0)/initrd
```

GRUB 會在下次啓動系統時，使用新參數。另一種方法是，使用YaST 開機載入器模組做此變更。將新參數添加至現有的行，由空格分開。

使用萬用字元以選取開機核心

特別是在開發或使用自訂核心時，您需要變更menu.lst 中的項目，或是編輯指令行以反映目前核心及initrd 的檔案名稱。若要簡化此程序，請使用萬用字元以動態更新GRUB 的核心清單。接著會將所有符合特定型式的核心影像都自動加入可開機影像的清單。請注意，未提供此功能的技術支援。

在menu.lst 中輸入其他的功能表項目以啓動萬用字元選項。爲了使用上的便利，所有的核心和initrd 影像必須具有共同的基礎名稱以及符合核心與其相關initrd 的識別碼。請思考以下設定：

```
initrd-default
initrd-test
vmlinuz-default
vmlinuz-test
```

在此狀況中，您可以在一個GRUB 組態中同時新增兩個開機影像。若要取得功能表項目linux-default 與linux-test，將需要menu.lst 中的下列項目：

```
title linux-*
  wildcard (hd0,4)/vmlinuz-*
  kernel (hd0,4)/vmlinuz-* root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-*
```

在此範例中，GRUB 會搜尋分割區(hd0,4) 以尋找符合萬用字元的項目。這些項目會用來產生新的GRUB 功能表項目。在前述的範例中，GRUB 的行爲方式將如同下列項目存在於menu.lst 一般：

```
title linux-default
  wildcard (hd0,4)/vmlinuz-default
  kernel (hd0,4)/vmlinuz-default root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-default
title linux-test
  wildcard (hd0,4)/vmlinuz-test
  kernel (hd0,4)/vmlinuz-test root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-test
```

如果所使用的檔案名稱不一致，或是如果其中一個已展開的檔案(例如，initrd 影像) 已遺失，此組態就會發生問題。

8.3.2 檔案device.map

檔案device.map會將GRUB設備名稱對應至Linux設備名稱。在包含IDE和SCSI硬碟的混合系統中，GRUB必須透過特定程序來判斷啓動順序，因為GRUB不會存取BIOS的開機順序資訊。GRUB會將此分析的結果儲存在檔案/boot/grub/device.map中。在BIOS中的啓動順序設定成IDE在SCSI前面的系統，檔案device.map會顯示如下：

```
(fd0) /dev/fd0
(hd0) /dev/hda
(hdl) /dev/sda
```

因為IDE、SCSI和其他硬碟的順序取決於各種因素，而且Linux無法識別對應，所以檔案device.map可以手動設定。如果您在開機時發生問題，請檢查此檔案中的順序是否與BIOS中的順序一致，然後在需要時使用節8.3.4，“GRUB外圍程序”中所描述的GRUB外圍程序來修改它。啓動Linux系統之後，檔案device.map可以透過YaST開機載入器或其他選擇的編輯器，永久性編輯。在手動變更檔案device.map之後，執行以下指令來重新安裝GRUB。此指令會造成檔案device.map重新載入，並以grub.conf列示的指令執行：

```
grub --batch < /etc/grub.conf
```

8.3.3 檔案/etc/grub.conf

除了menu.lst和device.map之外，第三個重要的GRUB組態檔是/etc/grub.conf。此檔案包含指令grub正確安裝開機載入器時，需要的參數和選項：

```
root (hd0,4)
  install /grub/stage1 d (hd0) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
quit
```

個別項目的意義如下：

root (hd0,4) 此指令告訴GRUB將以下指令套用至第一個硬碟的第一個邏輯分割區(開機檔案的位置)。

install 參數 指令grub應該與參數install一起執行。開機載入器的stage1應該安裝在第一個硬碟的MBR(/grub/stage1 d (hd0))。stage2應該載入記憶體位址0x8000(/grub/stage2 0x8000)。最後項目((hd0,4)/grub/menu.lst)告訴GRUB到什麼地方尋找功能表檔案。

8.3.4 GRUB 外圍程序

GRUB 實際上有兩個版本：當成開機載入器以及當成 `/usr/sbin/grub` 的一般 Linux 程式。此程式稱為 *GRUB 外圍程序*。將 GRUB 安裝成硬碟或磁片上的開機載入器，這種功能是以指令 `install` 和 `setup` 的形式，整合在 GRUB。Linux 載入時，可以在 GRUB 外圍程序使用此指令。

不過，指令 `setup` 和 `install` 也可以在 Linux 啟動前的開機程序中使用。這可利於修復無法再啟動的缺陷系統，因為開機載入器錯誤的組態檔，可以手動輸入參數，便得以解決。在開機程序期間手動輸入參數，對於測試新設定但可避免損壞原始系統，有很大的幫助。只要使用像 `menu.lst` 的語法，輸入實驗性組態檔就可以了。測試此項目的功能，無需變更現有的組態檔。例如，要測試新核心，輸入指令 `kernel` 以及新核心的路徑。如果啟動程序失敗，下次啟動時，可以繼續使用未受損的 `menu.lst`。同樣地，指令行介面也可以用來輸入正確的參數，即使 `menu.lst` 檔案錯誤，也可以啟動系統。正在執行的系統中，正確的參數可以輸入在 `menu.lst`，讓系統變成可永久開機。

GRUB 設備與 Linux 設備名稱的對應，只有將 GRUB 外圍程序當成 Linux 程式執行時，才會發生關係（輸入 `grub`，如節 8.3.2，“檔案 `device.map`” 所述一般）。基於此目的，程式會讀取檔案 `device.map`。如需更多資訊，請參閱節 8.3.2，“檔案 `device.map`”。

8.3.5 設定啟動密碼

即使作業系統啟動之前，GRUB 也可以存取檔案系統。沒有 `root` 許可權的使用者，在此時可以存取 Linux 系統中的檔案（這些檔案在系統啟動後，他們並無法存取）。要封鎖此類型的存取或者防止使用者啟動特定作業系統，可以設定啟動密碼。

按照以下方式，以使用者 `root` 的身份設定啟動密碼：

1. 在根提示時，輸入 `grub`。
2. 在 GRUB 外圍程序加密密碼：

```
grub> md5crypt
密碼：****
Encrypted:$1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

3. 將加密字串貼到檔案 `menu.lst` 的全域段落：

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password --md5 $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

現在GRUB 指令只可以在按(P) 並輸入密碼後的啓動提示時執行。不過，使用者仍然可以從開機功能表，啓動所有作業系統。

4. 要防止一或多個作業系統從開機功能表啓動，將項目lock 新增至沒有密碼便不可以啓動的menu.lst 每一個段落。例如：

```
title linux
kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
initrd (hd0,4)/initrd
lock
```

重新啓動系統並從開機功能表選取Linux 之後，會顯示以下錯誤：

錯誤 32：必須驗證

按(Enter) 進入功能表。然後按(P)，取得密碼提示。輸入密碼並按(Enter) 之後，選取的作業系統(本案例為Linux) 應該會啓動。

啓動密碼和開頭顯示畫面

如果您在GRUB 使用啓動密碼，將不會顯示一般的開頭顯示畫面。

8.4 使用YaST 設定開機載入程式

在SUSE LINUX 系統中設定開機載入程式最簡單的方法是使用YaST 模組。在「YaST 控制中心」選取‘系統’→‘開機載入程式組態’。系統目前的開機載入程式組態會顯示出來，供您進行必要的變更。請參閱圖形8.1, “使用YaST 設定開機載入程式”。

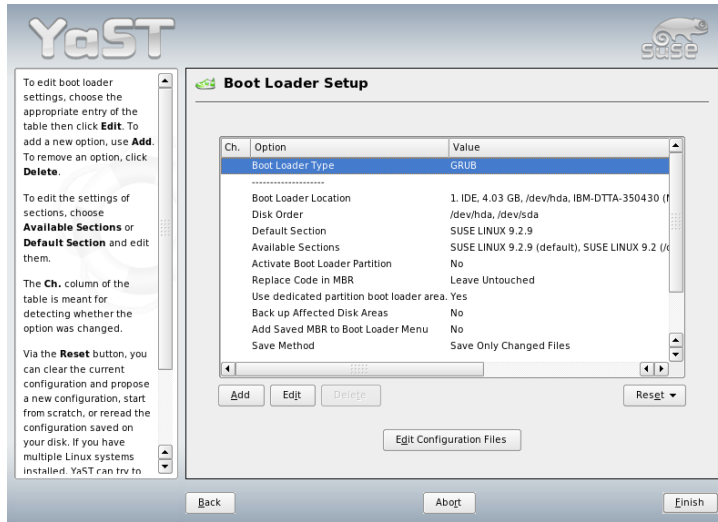


Figure 8.1: 使用YaST 設定開機載入程式

8.4.1 主視窗

列出組態資料的表格由三個欄位組成。在‘已變更’(在左邊)，有一些標誌標示中央欄位已經變更的選項。要加入選項，請按一下‘新增’。要變更現有選項的值，請先用滑鼠選取，接著按一下‘編輯’。如果您不想使用某個現有選項，請先選取再按一下‘刪除’。‘重設’提供下列選項：

建議新組態 產生新的組態建議。舊版Linux 或其他分割區中找到的其他作業系統都會列在開機功能表中，供您執行Linux 開機或使用舊版開機載入程式。後者會帶您進入第二個開機功能表。

從頭開始 供您從頭開始建立整個組態。將不會產生任何建議。

從磁碟重新讀取組態 如果您已經執行一些變更但對結果不滿意，請使用這個選項來重新載入目前的組態。

使用現有GRUB 功能表建議和合併 如果其他分割區中已經安裝另一個作業系統或舊版Linux，產生的功能表將含有表示新SUSE LINUX 的項目、表示其他系統的項目，以及舊開機載入程式功能表的所有項目。這個程序需要一點時間。如果使用LILO，則不會出現這種情況。

還原硬碟的MBR 寫回儲存在硬碟中的備份MBR。

請使用‘編輯組態檔’在編輯器中編輯相關的組態檔。要編輯檔案，請使用選擇欄位來載入。請按一下‘確定’來儲存變更。要離開開機載入程式組態，請按一下‘取消’。請按一下‘上一步’回到主視窗。

8.4.2 開機載入程式組態選項

使用YaST來執行組態會比直接編輯檔案更簡單。請選取選項接著按一下‘編輯’來開啓對話方塊，針對需求變更設定值。請按一下‘確定’來確認變更並回到主功能表，以編輯其他選項。可用的選項視使用的開機載入程式而定。下列清單介紹開機載入程式GRUB的一些選項：

開機載入程式類型 這個選項用來切換GRUB和LILO。請繼續進入其他對話方塊，來指定這個變更的執行方式。例如，將目前的GRUB組態轉換至相似的LILO組態。不過，如果沒有對應可用的選項，有些選項可能會遺失。您也可以從頭建立新組態，或是先產生組態再編輯提供的建議。

如果您在執行中的系統啓動開機載入程式組態，您可以從硬碟載入組態。如果您決定回到原始開機載入程式，可以使用最後一個選項來載入它的組態。不過，這只限於您未關閉開機載入程式模組才適用。

開機載入程式位置 使用這個對話方塊來定義開機載入程式的安裝位置：主要開機磁區(MBR)、開機分割區的開機磁區(如果有的話)、根分割區的開機磁區，或磁片。請使用‘其他’來指定不同位置。

硬碟順序 如果您的電腦有多個硬碟，請指定機器的BIOS設定中指定的硬碟開機順序。

預設區段 這個選項可以設定預設啓動的核心或作業系統。在超過時限時便會啓動所選的系統。在這個功能表中，可以使用‘編輯’按鈕來取得所有開機功能表項目的清單。請從清單選取一個項目，接著按一下‘設定為預設’。您現在仍可以使用‘編輯’來修改任何項目。

可用區段 現有的開機功能表項目列在主視窗中這個選項的下方。如果您選取這個選項接著按一下‘編輯’，將開啓一個和‘預設項目’完全相同的對話方塊。

啓用開機載入程式分割區 請使用這個選項來啓動其開機磁區含有開機載入程式的分割區，該分割區和含有儲存開機載入程式輔助功能檔案的目錄(/boot或根目錄/)的分割區無關。

置換MBR中的程式碼 如果GRUB已經安裝在MBR或者您要將系統安裝在新硬碟但又不想將GRUB安裝在MBR，請使用這個選項將標準開機程式碼還原至MBR。

備份檔案和部份硬碟內容 備份內容改變的硬碟區域。

新增儲存的MBR 至開機載入程式功能表

將儲存的MBR 新增至開機載入程式功能表。

請使用‘時限’來定義預設系統啓動前開機載入程式等候鍵盤輸入的秒數。其他一些選項可以使用‘新增’來指定。如需可用選項的詳細資訊，請參閱對應的手冊頁((grub(8) 或lilo(8)) 以及在<http://www.gnu.org/software/grub/manual/> 的線上說明文件。

8.5 解除安裝Linux 開機載入器

YaST 可以用來解除安裝Linux 開機載入器，並將MBR 還原回安裝Linux 前的狀態。安裝時，YaST 會自動建立原始MBR 的備份，然後在需要時還原，覆寫GRUB。

要解除安裝GRUB，請開啓YaST 開機載入器模組(‘系統’→‘開機載入器組態’)。在第一個對話方塊，選取‘重設’→‘還原硬碟的MBR’，然後使用‘完成’來結束對話方塊。在MBR 中，原給的MBR 資料會覆寫GRUB。

8.6 建立開機CD

如果使用開機管理員來啓動系統發生問題，或者開機管理員無法安裝在硬碟或磁片的MBR 上，也可以建立一張包含Linux 所有必要啓動檔案的開機CD。您的系統需要有一個CD 燒錄器。

要使用GRUB 建立可開機的CD-ROM 只需要一個特殊形式的`stage2`，它叫做`stage2_eltorito`，還可以選用自訂的`menu.lst`。不需要典型的檔案`stage1` 和`stage2`。

建立ISO 影像要使用的目錄，例如，您可以使用`cd /tmp` 和`mkdir iso`等指令。同時，使用`mkdir -p iso/boot/grub` 指令建立一個GRUB 子目錄。將檔案`stage2_eltorito` 複製到目錄`grub`：

```
cp /usr/lib/grub/stage2_eltorito iso/boot/grub
```

另外，將核心(`/boot/vmlinuz`)、`initrd` (`/boot/initrd`) 以及檔案`/boot/message` 複製到`iso/boot/`：


```
cp /boot/vmlinuz iso/boot/  
cp /boot/initrd iso/boot/  
cp /boot/message iso/boot/
```

若要讓它們可供GRUB使用，請將檔案menu.lst複製到iso/boot，然後調整路徑項目，使它們指向CD-ROM光碟機。方法是以CD-ROM光碟機的裝置名稱，也就是(cd)來取代路徑名稱中的硬碟裝置名稱，也就是格式中所列的(hd*)：

```
gfxmenu (cd)/boot/message  
timeout 8  
default 0  
  
title Linux  
    kernel (cd)/boot/vmlinuz root=/dev/hda5 vga=794 resume=/dev/hda1  
    splash=verbose showopts  
    initrd (cd)/boot/initrd
```

最後，使用以下指令建立ISO影像：

```
mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot \  
-boot-load-size 4 -boot-info-table -o grub.iso iso
```

使用您偏好的公用程式，將產生的檔案grub.iso燒錄至CD。

8.7 圖形化SUSE畫面

從SUSE LINUX 7.2開始，如果使用「vga={value?}」選項作為核心參數，則圖形化SUSE畫面會顯示在第一個主控台。如果您使用YaST來安裝，則會依照選取的解析度與顯示卡來自動啟動此選項。視需要，有三種方式可以停用SUSE畫面：

必要時停用SUSE畫面。 在指令行輸入指令echo 0 >/proc/splash以停用圖形式畫面。若要再次啟動它，請輸入echo 1 >/proc/splash。

預設停用SUSE畫面。 您可以新增核心參數splash=0到開機載入器的組態。章8, 開機載入器能提供更多有關這方面的資訊。但是，如果您想使用文字模式(較早版本的預設值)，則請設定vga=normal。

完全停用SUSE畫面。 編譯新的核心且關閉‘framebuffer支援’中的‘使用開機顯示畫面而非開機標幟’選項。

在核心中關閉framebuffer 支援將會自動關閉開機顯示畫面。如果您使用自訂核心來執行，SUSE 將不會為您的系統提供任何支援。

8.8 疑難排解

此章節列示使用GRUB 啓動時，經常遇到的問題以及可能解決方案的簡要說明。部份問題已經在「支援資料庫」中討論，網址是<http://portal.suse.de/sdb/en/index.html>。如果您的特殊問題未包含在此清單中，使用「支援資料庫」的搜尋對話方塊(網址是<https://portal.suse.com/PM/page/search.pm>)來搜尋像GRUB、boot 以及boot loader 的關鍵字。

GRUB 和XFS XFS 在分割區啓動區塊中，不會保留空間給stage1。因此，不要將XFS 分割區指定成開機載入器的位置。建立不是以XFS 格式化的個別啓動分割區，就可以解決此問題。

GRUB 和JFS 雖然技術上可行，不過GRUB 和JFS 的組合仍有問題。在此案例中，建立個別啓動分割區(/boot)並以Ext2 格式化。在此分割區安裝GRUB。

GRUB 報告GRUB 位置錯誤 GRUB 會在系統啓動時檢查連接硬碟的位置。有時候，BIOS 會傳回不一致的資訊，而且GRUB 會報告GRUB 位置錯誤。如果發生此狀況，使用LILO 或更新BIOS。如需關於LILO 的安裝、組態和維護的詳細資訊，請參閱「支援資料庫」中的關鍵字LILO。

如果Linux 安裝在其他硬碟上，而且未註冊在BIOS，GRUB 也會傳回此錯誤訊息。開機載入器的stage1 可以正確找到和載入，不過stage2 則找不到。將新磁碟註冊在BIOS 就可以解決此問題。

包含IDE 和SCSI 硬碟的系統不會啓動

安裝時，YaST 可能已經判斷硬碟的啓動順序錯誤(而且您尚未更正它)。例如，GRUB 可以將/dev/hda 當成hd0 而將/dev/sda 當成hd1，即使BIOS 中的啓動順序是其他方式(SCSI 在IDE 前面)。

發生此狀況，在啓動程序時，透過GRUB 指令行的協助來更正硬碟。系統啓動之後，編輯檔案device.map 來永久套用新的對應。然後檢查檔案/boot/grub/menu.lst 和/boot/grub/device.map 中的GRUB 設備名稱，然後使用以下指令，重新安裝開機載入器：

```
grub --batch < /etc/grub.conf
```

從第二顆硬碟啟動Windows 有些作業系統，例如Windows，只可以從第一個硬碟啟動。在第一個硬碟以外的硬碟安裝類似作業系統時，會影響個別功能表項目的邏輯變更。

```
...
title windows
map (hd0) (hd1)
map (hd1) (hd0)
chainloader(hd1,0)+1
...
```

在此範例中，Windows 是從第二個硬碟啟動。基於此目的，硬碟的邏輯順序是使用map 變更的。此變更不會影響GRUB 功能表檔案中的邏輯。因此，必須為chainloader 指定第二個硬碟。

8.9 若需詳細資訊

如需關於GRUB 更進一步的資訊，請參閱<http://www.gnu.org/software/grub/>。如果您的機器上裝有texinfo，輸入info grub 可以檢視外圍程序中的GRUB 資訊頁面。您也可以在此「支援資料庫」搜尋關鍵字“GRUB”，取得關於特殊問題的相關資訊，網址是<http://portal.suse.de/sdb/en/index.html>。

Linux 核心

核心管理每個Linux系統的硬體，以便應用於各種不同的程序。雖然本章中提供的資訊不能讓您成為核心高手，但是還是能夠學習到如何執行核心更新，以及編譯和安裝自訂核心的方法。如果依照本章中的說明，以前的核心可以繼續作用，而且必要時也可以啓動。

9.1	核心更新	164
9.2	核心來源	164
9.3	核心組態	164
9.4	核心模組	166
9.5	編譯核心	168
9.6	安裝核心	168
9.7	編譯後清理硬碟	169

安裝於 `/boot` 目錄的核心是為廣泛的硬體設定的。一般而言，除非您要測試實驗性的功能和驅動程式，否則不需要編譯自訂核心。

通常已安裝核心的行為可以透過核心參數來修改。例如，參數 `desktop` 為設定程式較短的時間配量，在主觀上會造成系統加速。核心文件中提供的資訊位於目錄 `/usr/src/linux/Documentation` 中，這是假設已安裝套件 `kernel-source`。

核心隨附數個 `Makefiles` 來自動化程序。選取硬體設定及其他核心功能。因為您需要熟知電腦系統才能選取正確的設定，所以建議您先試著修改現有的工作組態檔。

9.1 核心更新

若要安裝正式的 SUSE 更新核心，請使用 YaST 的線上更新功能。在核心更新後，必須重新啟動系統，因為正在執行中的舊核心將會找不到適當的模組來提供所需的功能。如需關於 YaST 的詳細資訊，請參閱節 2.2.3, “YaST 線上更新” 的線上更新。

當執行更新時，會出現快顯功能表以說明所有需要的動作。遵循這些指令以維護一致的系統。

9.2 核心來源

若要建立核心，必須安裝 `kernel-source` 套件。其他的套件，如 C 編譯器 (`gcc` 套件)，GNU `binutils` (`binutils` 套件) 和 C 編譯器所包含的檔案 (`glibc-devel` 套件)，YaST 會自動選取它們進行安裝，而且也必須安裝它們。

安裝後，核心來源的位置在 `/usr/src/linux-<kernel-version>`。如果您想要嘗試不同的核心，請將這些核心在不同的子目錄中解除壓縮，然後建立到目前核心來源的符號連結。由於軟體套件所需依靠的來源位於 `/usr/src/linux`，因此請保持此目錄至目前核心來源的符號連結。YaST 會自動完成這個動作。

9.3 核心組態

目前核心的組態儲存於檔案 `/proc/config.gz`。若要修改此組態，請以 `root` 身份到 `/usr/src/linux` 目錄，然後執行下列指令：

```
zcat /proc/config.gz > .config
make oldconfig
```

`make oldconfig` 指令使用 `/usr/src/linux/.config` 檔案，做為目前核心組態的樣板。詢問您目前核心來源是否有任何新選項。如果 `.config` 檔案不存在，會使用核心來源內包含的預設組態。

在此無法詳細涵蓋核心的組態選項。請儘量使用核心組態中提供的各種說明內容。最新的核心文件通常位於 `/usr/src/linux/Documentation`。

9.3.1 指令行上的組態

若要設定核心，請變更到 `/usr/src/linux` 並輸入 `make config` 指令。選擇您想要核心支援的功能。通常有二到三個選項：`(Y)`、`(N)` 和 `(M)`。`(M)` 表示此設備不應該直接編譯到核心，而是要當作模組載入。開機系統所需的驅動程式必須使用 `(Y)` 與核心整合。按 `(Enter)` 確認從 `.config` 檔案讀取的預設設定。按其他任何一鍵，查看相關選項的簡短說明文字。

9.3.2 文字模式中的組態

`menuconfig` 是設定核心較便利的方式。如有需要，請使用 YaST 安裝 `ncurses-devel`。使用 `make menuconfig` 指令啟動核心組態。

對於組態中的細微變更，您不需要瀏覽所有的問題。而是使用功能表直接存取特定章節。預設設定從 `.config` 檔案載入。若要載入不同的組態，請選取‘載入其他組態檔案’並輸入檔案名稱。

9.3.3 X Window 系統中的組態

如果您已安裝和設定 X Window 系統 (`xorg-x11` 套件) 與 QT 開發套件 (`qt3-devel`)，可以使用 `make xconfig` 指令存取組態的圖形使用者介面。如果您不是以 `root` 登入 X Window 系統，請輸入 `su` 指令，以取得 `root-shell` 的顯示權限。預設設定從 `.config` 檔案載入。因為使用 `make xconfig` 設定的組態維護可能不如其他組態，所以使用此組態方法後，請執行 `make oldconfig` 指令。

9.4 核心模組

有各式各樣的PC硬體元件。若要正確使用此硬體，作業系統(在Linux中是核心)的「驅動程式」必須能存取該硬體。基本上有兩種方式將驅動程式整合到系統：

- 驅動程式可以直接編譯到核心。這類核心(「一套件」)是指*monolithic*核心。有些驅動程式僅能以此形式使用。
- 視需要將驅動程式載入至核心。在這種情況下，核心被稱為*modularized*核心。這種核心的好處是，僅會載入所需的驅動程式，因此核心不會包含不需要的物件。

要編譯到核心的那些驅動程式及載入為執行階段的模組，會定義於核心組態中。基本上，系統開機時不需要的元件應該要建立為模組。如此可確保核心不會變得太大而讓BIOS或開機載入器無法載入。ext2的驅動程式、以SCSI為基礎系統的SCSI驅動程式、以及類似的驅動程式，應該編譯到核心中。相反地，啓動電腦系統時不需要的項目，如isofs、msdos、或sound，應該建立為模組。

即使需要啓動系統的驅動程式可能是以模組建立。在此例中，起始ramdisk用於在啓動時載入這些模組。

核心模組位於/lib/modules/<version>；version代表目前的核心版本。

9.4.1 使用hwinfo的說明進行硬體偵測

hwinfo可以偵測系統上的硬體，以及選取執行此硬體所需的驅動程式。使用hwinfo --help取得關於此指令的簡介。例如，如果您需要有關SCSI設備的資訊，請使用hwinfo --scsi指令。您也可以從硬體資訊模組的YaST中取得所有的資訊。

9.4.2 處理模組

將模組載入核心的公用程式可在module-init-tools套件中取得。下列是可用的指令：

insmod insmod 在 `/lib/modules/<version>` 的子目錄中搜尋後會載入要求的模組。不過，使用 `modprobe` 會比 `insmod` 好，因為 `modprobe` 也會檢查模組的依存度。

rmmmod 卸載要求的模組。只有不再需要此模組時才能卸載模組。例如，還裝載 CD 時無法卸載 `isofs` 模組。

depmod 在定義所有模組之依存度的 `/lib/modules/<version>` 中建立 `modules.dep` 檔案。這是必要的做法，可確保所有依存的模組都與選取的模組一起載入。系統啟動後，如果檔案不存在，就會建立該檔案。

modprobe 載入或卸載指定模組時，應考慮該模組的依存度。此指令非常強大，可以用於執行許多動作，包括成功載入模組前，查探特定類型的所有模組。與 `insmod` 不同的是，`modprobe` 會檢查 `/etc/modprobe.conf`，讓它成為載入模組的慣用方法。如需有關此主題的詳細資訊，請參閱相關的 `man` 頁面。

lsmod 顯示目前載入哪些模組以及正在使用這些載入模組的其他模組數目。核心精靈啟動的模組尾部加上 `autoclean` 標籤。此標籤表示這些模組在達到其閒置時間限制時會自動移除。

modinfo 顯示模組資訊。因為此資訊是從模組本身擷取的，所以只會顯示驅動程式開發人員建立的資訊。該資訊可能包括了作者、描述、授權、模組參數、依存度以及別名。

9.4.3 /etc/modprobe.conf

模組的載入會受到 `/etc/modprobe.conf` 與 `/etc/modprobe.conf.local` 檔案和 `/etc/modprobe.d` 目錄的影響。請參閱 `man modprobe.conf`。直接存取硬體的模組參數必須輸入到此檔案。這類模組，例如 CD-ROM 驅動程式或網路驅動程式，可能需要系統特定的選項。此處使用的參數，描述於核心來源中。安裝 `kernel-source` 套件以及參閱 `/usr/src/linux/Documentation` 目錄中的文件。

9.4.4 Kmod — 核心模組載入器

核心模組載入器是使用模組的最佳方式。`Kmod` 執行背景監控，確定核心中需要相關功能時，`modprobe` 就會載入必要模組。

若要使用 `Kmod`，請啟動核心組態中的‘核心模組載入器’(`CONFIG_KMOD`) 選項。`Kmod` 無法自動卸載模組。就今日的 RAM 容量觀點而言，潛在節省的記憶體也許微不足道。

9.5 編譯核心

► x86, AMD64, EM64T

建議編譯“bzImage”。這樣通常可避免發生核心變得過大的問題，而且如果選取太多功能並建立“zImage”時，很容易發生這種情形。接著您就會看到kernel too big 或System is too big 的錯誤訊息。◀

如節9.3, “核心組態” 中所述以自訂核心組態後，輸入以下文字開始編譯(記得要先對/usr/src/linux 目錄進行變更)：

```
make clean
make bzImage
```

這兩個指令可以輸入為一個指令行：

```
make clean bzImage
```

成功編譯後，壓縮的核心位於/usr/src/linux/arch/<arch>/boot。核心影像(包含核心的檔案)稱為bzImage。

如果找不到此檔案，進行核心編譯時可能會發生錯誤。在Bash 外圍程序中，輸入以下指令再次啟動核心編譯，並將輸出寫入到kernel.out 檔案：

```
make bzImage V=1 2>&1 | tee kernel.out
```

如果已設定核心的部份載入為模組，請啟動模組編譯。使用make modules 執行這個動作。

9.6 安裝核心

編譯核心後，必須安裝該核心才能開機。核心必須安裝在/boot 目錄中。使用下列指令來執行此動作：

```
INSTALL_PATH=/boot make install
```

現在需要安裝已編譯的模組。輸入make modules_install 將模組複製到/lib/modules/<version> 中正確的目標目錄。如果核心版本相同，會覆寫舊的模組。不過，原始模組可以從CD 與核心一起重新安裝。

若要避免發生不可預期的效果，請確定模組(其功能可直接編譯到核心)已從/lib/modules/<version> 中移除。這就是為什麼「強烈」建議沒有經驗的使用者不要編譯核心的原因之一。

若要啓用GRUB來啓動舊的核心(現在是/boot/vmlinuz.old)，請新增Linux.old 標籤做爲/boot/grub/menu.lst 檔案中的開機影像。這個程序在章8, 開機載入器中有詳細說明。不需要重新安裝GRUB。

/boot/System.map 檔案包含模組所需的核音符號，以確保可以成功啓動核心功能。此檔案視目前的核心而定。因此，一旦編譯和安裝核心後，請複製/usr/src/linux/System.map 到目錄/boot。每次編譯核心時就會重新產生該檔案。如果看到System.map 不符合目前核心的錯誤訊息，很可能是因爲您在編譯核心後，忘記複製System.map 到/boot。

9.7 編譯後清理硬碟

如果硬碟空間不足，請使用/usr/src/linux目錄中的make clean 指令，以刪除核心編譯期間產生的物件檔。如果硬碟空間足夠且打算定期重新設定核心，可以略過這項作業。重新編譯核心的速度較快，因爲僅會重新編譯受到變更影響的部分。

SUSE LINUX 的特殊功能

本章節會提供關於軟體套件、虛擬主控台及鍵盤配置等資訊。接著是關於語言與國家專用設定的章節(I18N 與L10N)。

10.1	特殊軟體套件的資訊	172
10.2	虛擬主控台	179
10.3	鍵盤配置	179
10.4	語言與國家專用的設定	180

10.1 特殊軟體套件的資訊

10.1.1 Bash 套件與/etc/profile

下列是在使用Bash 做為登入外圍程式時，所讀取的全部init 檔案清單。Bash 會以它們顯示在清單中的順序來處理。

1. /etc/profile
2. ~/.profile
3. /etc/bash.bashrc
4. ~/.bashrc

使用者可以在~/.profile 或~/.bashrc 中建立個人設定項目。爲了要確保這些檔案能正確的處理，您必須將基本設定從/etc/skel/.profile 或/etc/skel/.bashrc 中複製至使用者的主目錄。建議您在更新後，從/etc/skel 複製設定。請執行下列的外圍程式指令，以避免遺失您調整過的設定。

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

然後個人的調整設定需要從*.old 檔案再複製回去。

10.1.2 cron 套件

cron 表格現在位於/var/cron/tabs。/etc/crontab 做為整個系統的cron 表格。請在時間表格之後，輸入應該直接執行指令的使用者名稱。在範例10.1, "/etc/crontab 中的項目範例" 中，則是輸入root。位於/etc/cron.d 的套件專用表格有相同的格式。請參閱man cron。

Example 10.1: /etc/crontab 中的項目範例

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

`/etc/crontab` 無法用 `crontab -e` 來處理。必須直接載入編輯器中進行修改，然後儲存。

有些套件會將外圍程式程序檔安裝至 `/etc/cron.hourly`、`/etc/cron.daily`、`/etc/cron.weekly` 及 `/etc/cron.monthly` 等目錄中，其指令會由 `/usr/lib/cron/run-crons` 所控制。`/usr/lib/cron/run-crons` 每隔 15 分鐘會從主表格 (`/etc/crontab`) 執行一次。這會保證被忽略的程序可以在適當的時間執行。

每日的系統維護工作因為明確的理由，已經配送到不同的程序檔。它們包含在 `aaa_base` 套件中。例如，`/etc/cron.daily` 中有 `backup-rpmdb`、`clean-tmp` 或 `clean-vi` 等元件。

10.1.3 記錄檔：logrotate 套件

某些系統服務 (*daemon*) 以及核心本身，會定期將系統狀態與特定事件記錄到記錄檔中。這樣管理員可以定期在某個時間點檢查系統的狀態、找出錯誤或有問題的功能，並且用精確的方式來排除它們。這些記錄檔通常以 FHS 所指定的方式儲存於 `/var/log`，而且會日益增大。logrotate 套件有助於控制這些檔案增大的方式。

組態

使用 `/etc/logrotate.conf` 檔案來設定 logrotate。特別是，`include` 設定主要是設定其他要讀取的檔案。SUSE LINUX 可以確保個別的套件會將檔案安裝在 `/etc/logrotate.d` (例如，`syslog` 或 `yast`)。

Example 10.2: /etc/logrotate.conf 的範例

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
```

```
# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#   monthly
#   create 0664 root utmp
#   rotate 1
#}

# system-specific logs may be also be configured here.
```

logrotate 是透過cron 來控制，並且每日是經由/etc/cron.daily/logrotate 來呼叫。

create 選項會讀取/etc/permissions* 中由管理員所做的所有設定。請確定個人的修改不會造成衝突。

10.1.4 Man 頁面

某些GNU 應用程式(例如tar) 不再支援man 頁面。針對這些指令，請使用--help 選項來取得快速概觀或資訊頁面，這些都會提供更深入詳盡的說明。info 是GNU 的超文字系統。您可以輸入info info 來讀取此系統的介紹。您可以輸入emacs -f info 或直接在主控台中使用info，以便使用Emacs 檢視資訊頁面。您也可以使用tinfo、xinfo 或SUSE 說明系統檢視info 頁面。

10.1.5 locate 指令

關於快速尋找檔案的locate 並不包含在安裝軟體的標準範圍中。若有需要，請安裝套件(find-locate)。updatedb 程序會在每晚自行啓動，或在啓動系統後的15分鐘左右啓動。

10.1.6 ulimit 指令

利用ulimit (使用者限制) 指令，您可以限制系統資源的使用，並顯示這些限制。ulimit 對於限制應用程式可用的記憶體特別有用。利用它，可以避免應用程式使用過多的記憶體空間，使用過多的記憶體空間可能會導致系統暫停。ulimit 可以搭配多種選項來使用。若要限制記憶體的使用，請利用表格10.1, "ulimit：設定使用者的資源" 中所列的選項。

Table 10.1: ulimit：設定使用者的資源

-m	實體記憶體的最大大小
-v	虛擬記憶體的最大大小
-s	堆疊的最大大小
-c	核心檔案的最大大小
-a	限制集的顯示

您可以在 `/etc/profile` 中做整個系統的設定。在此可建立核心檔，以供程式設計人員除錯之用。一般使用者無法增加系統管理員在 `/etc/profile` 中所指定的值，但可以在自己的 `~/.bashrc` 中建立特殊的設定項目。

Example 10.3: ulimit：在 `~/.bashrc` 中的設定

```
# Limits of physical memory:
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

記憶體的單位必須為KB。如需詳細資訊，請參閱 `man bash`。

並非所有的外圍程序都支援 `ulimit` 指令。在您根據這些限制的內含設定時，PAM (例如 `pam_limits`) 會提供全面的調整設定。

10.1.7 free 指令

如果您的目標是要找出目前使用了多少RAM的話，則是稍微誤解 `free` 指令的用途。您可以在 `/proc/meminfo` 找到相關的資訊。近來，使用Linux之類新式作業系統的使用者，真的不太需要擔心記憶體的問題。可用的RAM的概念要回溯到聯合記憶體管理的年代之前。記憶體要物盡其用的口號非常適用於Linux。所以，Linux一直致力於平衡快取，而不允許有剩餘或未使用的記憶體。

基本上，核心不會有任何應用程式或使用者資料的直接知識。相反地，它會在頁面快取中管理應用程式與使用者資料。如果記憶體不足，部份的記憶體資料會寫入交換分割區或檔案中，這部份一開始就能用mmap指令的協助來讀取(請參閱man mmap)。

此外，核心也可以有其它的快取，例如slab快取，網路存取的快取資料會儲存於此處。這可以解釋/proc/meminfo中計數器之間的不同。它們大部分但非全部都可以透過/proc/slabinfo來存取。

10.1.8 /etc/resolv.conf 檔案

領域名稱解析會透過/etc/resolv.conf檔案來處理。請參照章24, 領域名稱系統。

只有/sbin/modify_resolvconf程序檔可以更新此檔案，其它的程式沒有權限可以直接修改/etc/resolv.conf。能保證系統的網路組態與相關檔案都保持在一致狀態的唯一方法，就是強制執行此規則。

10.1.9 GNU Emacs 的設定

GNU Emacs是個複雜的工作環境。更多相關資訊可在<http://www.gnu.org/software/emacs/>取得。以下幾個小節包含在GNU Emacs啟動時組態檔案的處理情形。

啟動時，Emacs會讀取多個檔案，包含使用者、系統管理員與供應商的設定，以及取得自訂或預設組態的設定。起始化檔案~/.emacs會從/etc/skel安裝至個別使用者的主目錄。emacs接著會讀取/etc/skel/.gnu-emacs檔案。如果要自訂程式，請將.gnu-emacs複製到主目錄(利用cp/etc/skel/.gnu-emacs ~/.gnu-emacs指令)，並依照您的需求來設定。

.gnu-emacs定義~/.gnu-emacs-custom檔案為自訂檔案。如果使用者是使用自訂選項來進行設定，則會將它們儲存至~/.gnu-emacs-custom中。

透過SUSE LINUX，emacs套件可將site-start.el檔案安裝至/usr/share/emacs/site-lisp目錄中。site-start.el檔案會在起始化檔案~/.emacs前載入。此外，site-start.el會確保那些以Emacs附加套件來散佈的特定組態檔案皆能自動載入，例如psgml。此類型的組態檔案也位於/usr/share/emacs/site-lisp中，並且會以suse-start-為開頭。本機系統管理員可在default.el中指定整個系統的設定。

有關這些檔案的詳細資訊可在Init File下的Emacs資訊檔案中取得：info:/emacs/InitFile。關於如何關閉這些檔案的載入(若有需要)的資訊，也可在此取得。

Emacs的元件分成數個套件：

- emacs 基本套件。
- emacs-x11 (通常會安裝)：使用 X11 支援的程式。
- emacs-nox：不使用 X11 支援的程式。
- emacs-info：線上的info 格式文件。
- emacs-el：以Emacs Lisp 編寫的未編譯文件庫檔案。執行期間用不到這類檔案。
- 需要時可加以安裝的多個附加套件：emacs-auctex (用於LaTeX)、psgml (用於SGML 與XML)、gnuserv (用於用戶端和伺服器操作) 以及其他等等。

10.1.10 vi 的概略介紹

許多的系統管理工作及程式設計仍然會用到文字編輯器。在Unix 的世界中，vi 作為編輯器的表現十分突出，因為它提供了便利的編輯功能，而且支援滑鼠因而較其他編輯器更符合人體工學。

運作模式

基本上，vi 使用三種運作模式：插入模式、指令模式及延伸模式。按鍵會因為模式而有不同的功能。啟動時，vi 通常設定為「指令」模式。第一件要學習的事就是如何切換這些模式：

指令模式切換到插入模式 方法有很多種，包括輸入ⓐ 為附加、ⓑ 為插入或ⓒ 為在目前的行中插入新行。

插入模式切換到指令模式 按(ESC) 鍵便可離開「插入」模式。vi 是無法在「插入」模式中終止的，因此習慣按(ESC) 鍵是很重要的。

指令模式切換到延伸模式 vi 的「延伸」模式可以藉由輸入冒號(:) 來啟動。延伸或ex 模式就像是個獨立的命令行編輯器，可用來處理各種簡單與更複雜的工作。

延伸模式切換到指令模式 在延伸模式中執行完指令後，編輯器會自動回到指令模式。如果您決定不要執行「延伸」模式中的任何指令，請用(←) 鍵刪除冒號。編輯器便會回到指令模式。

您無法直接從插入模式切換到延伸模式，而不先切換到指令模式。

vi 像其他的編輯器一樣都有自己的終止程式的程序。您無法在「插入」模式中終止 vi。首先，按 **(Esc)** 鍵離開插入模式。然後，您會有兩種選項：

1. 不儲存就離開：若要終止編輯器而不儲存變更，請在「指令」模式中輸入 **(:)(q)!**。驚歎號 **!** 會造成 vi 忽略任何的變更。
2. 儲存並離開：有許多方法可以儲存您的變更並終止編輯器。在「指令」模式中，使用 **(Z)(Z)**。若要從「延伸」模式離開程式並儲存所有的變更，請輸入 **(:)(w)(q)**。在延伸模式中，**(w)** 代表“寫入”，**(q)** 代表“結束”。

使用 vi

vi 可以用來做為一般編輯器。在「插入」模式中，輸入文字，然後用 **(←)** 鍵與 **(Del)** 鍵刪除文字。使用方向鍵來移動游標。

不過，這些控制鍵常會造成問題，因為有很多種終端機使用特殊鍵碼。這就是指令模式派上用場之處。按 **(Esc)** 鍵從插入模式切換到指令模式。在指令模式中，請用 **(H)**、**(J)**、**(K)** 及 **(L)** 等鍵來移動游標。這些鍵有下列功能：

- (H)** 向左移動一個字元
- (J)** 向下移動一行
- (K)** 向上移動一行
- (L)** 向右移動一個字元

指令模式中的指令都允許不同的變化。若要執行一個指令數次，只要在輸入實際的指令之前，輸入要重複的次數即可。例如，輸入 **(5)(L)** 來將游標向左移動五個字元。

若需詳細資訊

vi 支援許多指令。它讓您可以使用巨集、捷徑、具名緩衝區及許多其他有用的功能。不同選項的詳細說明已超出本手冊的範圍。SUSE LINUX 會隨附 vim (加強的 vi)，也就是 vi 的加強版。有數個此應用程式的資訊來源：

- vimtutor 是 vim 的互動式教學課程。
- 在 vim 中，請輸入 **:help** 指令以取得很多的說明主題。

- 關於vim 的書籍可以在線上取得，網址為<http://www.truth.sk/vim/vimbook-OPL.pdf>。
- vim 計劃的網頁位於<http://www.vim.org>，網站中會有所有的新聞、郵件清單及其他的文件。
- 網際網路也有不少 vim 來源：<http://www.selflinux.org/selflinux/html/vim.html>、<http://www.linuxgazette.com/node/view/9039>、http://www.apmaths.uwo.ca/~xli/vim/vim_tutorial.html。請參閱<http://linux-universe.com/HOWTO/Vim-HOWTO/vim-tutorial.html> 取得更多有關教學課程的連結。

VIM 授權

vim 是「慈善軟體」，意即作者不索取任何的軟體費用，但鼓勵您以捐款贊助，來支持非營利的專案計畫。此計劃懇求您幫助烏干達的貧困孩童。更多的資訊可在線上取得，網址為<http://iccf-holland.org/index.html>、<http://www.vim.org/iccf/> 及<http://www.iccf.nl/>。

10.2 虛擬主控台

Linux 是多重使用者及多工的作業系統。這些功能的優點即使在獨立的個人電腦系統中一樣令人讚賞。在文字模式中，有六個虛擬主控台可用。請使用 **(Alt)-(F1)** 到 **(Alt)-(F6)** 這些鍵來切換虛擬主控台。第七個主控台會保留給 X。您可以修改 `/etc/inittab` 中的檔案來指定較多或較少的主控台。

若要在不關閉主控台的情況下，從 X 切換到主控台，請使用 **(Ctrl)-(Alt)-(F1)** 到 **(Ctrl)-(Alt)-(F6)** 這些鍵。之後按 **(Alt)-(F7)** 便可返回 X。

10.3 鍵盤配置

若要標準化程式的鍵盤配置，請變更下列的檔案：

```
/etc/inputrc
/usr/X11R6/lib/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/X11R6/lib/X11/app-defaults/XTerm
/usr/share/emacs/<VERSION>/site-lisp/term/*.el
```

這些變更僅會影響使用terminfo項目的應用程式，或直接變更其組態檔的應用程式(vi、less等等)。未隨附於SUSE LINUX的應用程式必須相容於這些預設值。

在X中，可以使用(Ctrl)-(Shift)(右邊)來存取組合鍵(多重鍵)。請參閱/usr/X11R6/lib/X11/Xmodmap中的對應項目。

可在“X鍵盤延伸程式”(XKB)上進行進一步的設定。GNOME(gswitchit)以及KDE(kxkb)桌面環境也會使用此延伸程式。有關XKB資訊可在/etc/X11/xkb/README檔案中取得，也可以從列於該檔案中的文件取得。

關於中文、日文以及韓文(CJK)輸入的詳細資訊，可在Mike Fabian網頁中取得：<http://www.suse.de/~mfabian/suse-cjk/input.html>。

10.4 語言與國家專用的設定

SUSE LINUX在很大程度上是國際化的軟體，而且能以彈性的方式針對當地的需求進行修改。換句話說，國際化(I18N)允許特定的當地語系化(L10N)。I18N與L10N這兩個縮寫是取首尾兩個字母，兩字母中間再加上省略的字母數目。

設定位於/etc/sysconfig/language中所定義的LC_變數。設定範圍除了本地語言支援外，還包括訊息(語言)、字元集、排序順序、時間和日期、數字及貨幣等類別。每種類別都可以用自己的變數來直接定義，或用在language檔案中的主要變數來間接定義(請參閱man locale手冊頁)。

RC_LC_MESSAGES、RC_LC_CTYPE、RC_LC_COLLATE、RC_LC_TIME、RC_LC_NUMERIC、RC_LC_MONETARY

這些變數會傳送到外圍程序，但不會包含RC_字首，並會決定上述的類別。相關檔案會列於下面。目前的設定可以用locale指令來顯示。

RC_LC_ALL 此變數(如果設定)會覆寫上述所提到的變數值。

RC.LANG 如果没有設定上面的變數，則此為備用變數。依預設，SUSE LINUX 僅會設定RC.LANG。這讓使用者更容易輸入自己的值。

ROOT_USES_LANG 有yes 或no 兩個變數。如果設為no 的話，則root 永遠可在POSIX 環境中作業。

其他的變數可以透過YaST sysconfig 編輯器來設定。這樣的變數值中包含語言碼、國碼、編碼及修飾元。個别的元件會以特定的字元來連接：

```
LANG=<language>[[_<COUNTRY>].<Encoding>[@<Modifier>]]
```

10.4.1 一些範例

您必須將語言與國碼一起設定。語言設定要依循標準的ISO 639 (<http://www.evertype.com/standards/iso639/iso639-en.html> 與 <http://www.loc.gov/standards/iso639-2/>)。國碼會在ISO 3166 中列出(http://www.din.de/gremien/nas/nabd/iso3166ma/codlstpl/en_listpl.html)。只有設定那些可以在/usr/lib/locale 中找到的可用描述檔案的值，才會有意義。您可以用localedef 指令從/usr/share/i18n 中的檔案建立其他描述檔；描述檔屬於glibc-i18ndata 套件的一部份。en_US.UTF-8 (針對美式英文)的描述檔可以用以下指令建立：

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

LANG=en_US.UTF-8 如果安裝期間選擇英文的話，則此為預設設定。如果您選擇了其他語言，則仍然可以使用該語言，但會以UTF-8 做為字元編碼。

LANG=en_US.ISO-8859-1 這會將語言設成英文、國家設成美國、字元集設成ISO-8859-1。此字元集並不支援歐元符號，但有時對於尚未支援UTF-8 的程式卻非常實用。然後，有些程式將會評估定義字元集的(此例為ISO-8859-1)的字串，像是Emacs。

LANG=en_IE@euro 上方範例在語言設定中明確包括歐元符號。嚴格來說，這個設定現在已不再使用，因為UTF-8 也涵蓋歐元符號。它只有在應用程式不支援UTF-8 而只支援ISO-8859-15 時才有用。

SuSEconfig 會讀取/etc/sysconfig/language 中的變數，並將必要的變更寫入/etc/SuSEconfig/profile 與/etc/SuSEconfig/csh.cshrc 中。/etc/profile 將讀取/etc/SuSEconfig/profile，或將之做為來

源。/etc/SuSEconfig/csh.cshrc 則是/etc/csh.cshrc 的來源。這讓這些設定能在整個系統中使用。

使用者可以適當地編輯自己的~/.bashrc 來覆寫系統預設值。例如，若不要讓整個系統的程式訊息皆使用en_US時，請加入LC_MESSAGES=es_ES，便會改用西班牙文來顯示訊息。

10.4.2 語言支援的設定

按照規定，在訊息類別中的檔案僅會儲存於對應的語言目錄中(像是en)，以便有備用可用。如果您將LANG設為en_US，而且/usr/share/locale/en_US/LC_MESSAGES 中的訊息檔案不存在的話，則它會回到/usr/share/locale/en/LC_MESSAGES 中。

您也可以定義備用鍊，例如，不列塔尼文之於法文，或是加里斯亞文之於西班牙文之於葡萄牙文：

```
LANGUAGE="br_FR:fr_FR"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

如有需要，請改用挪威文變體*nynorsk* 與*bokmal* (讓其他備用為否)：

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

或

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

請注意，在挪威文中，會以不同方式處理LC_TIME。

可能的問題

不能辨識千位數逗號。LANG可能會設為en，但glibc所用的描述卻位於/usr/share/lib/en_US/LC_NUMERIC 中。例如，LC_NUMERIC必須設為en_US。

若需詳細資訊

- *The GNU C Library Reference Manual* 的“Locales and Internationalization”一章；包含在glibc-info 中。
- Markus Kuhn 所寫的*UTF-8 and Unicode FAQ for Unix/Linux*，目前網址如下：<http://www.cl.cam.ac.uk/~mgk25/unicode.html>。
- *Unicode-Howto*，Bruno Haible 著：`file:/usr/share/doc/howto/en/html/Unicode-HOWTO.html`。

X Window System

X Window System (X11) 是UNIX 中既成現實標準的圖形化使用者介面。X 採網路架構，可讓應用程式在一個主機上啓動而在透過任何種類的網路(LAN 或網際網路) 連線的其他主機上顯示。

本章節中描述設定、最佳化的可行方式、在SUSE LINUX 下使用字型的背景資訊以及OpenGL 與3D 的組態。

11.1	使用SaX2 來設定X11	184
11.2	X 組態最佳化	192
11.3	安裝與設定字型	197
11.4	OpenGL — 3D 組態	202

11.1 使用SaX2 來設定X11

圖形使用者介面(或X 伺服器)可處理軟硬體間的通訊。如KDE 與GNOME 等桌面以及許多視窗管理員，都可以使用X 伺服器與使用者互動。

圖形使用者介面會在安裝期間初次設定。之後若要變更該設定，請執行SaX2。會儲存目前的設定，而您可以隨時重設。目前的值會顯示出來以供修改：螢幕解析度、色彩深度、更新速率與顯示器的廠商與類型(如果自動偵測到的話)。

如果剛剛安裝新的圖形介面卡，就會出現一個小對話方塊詢問您是否要啓動圖形介面卡的3D 加速功能。按一下‘編輯’。輸入與顯示設備的組態工具SaX2，就會在另一個視窗中啓動。此視窗顯示於圖形11.1, “SaX2 的主要視窗” 中。

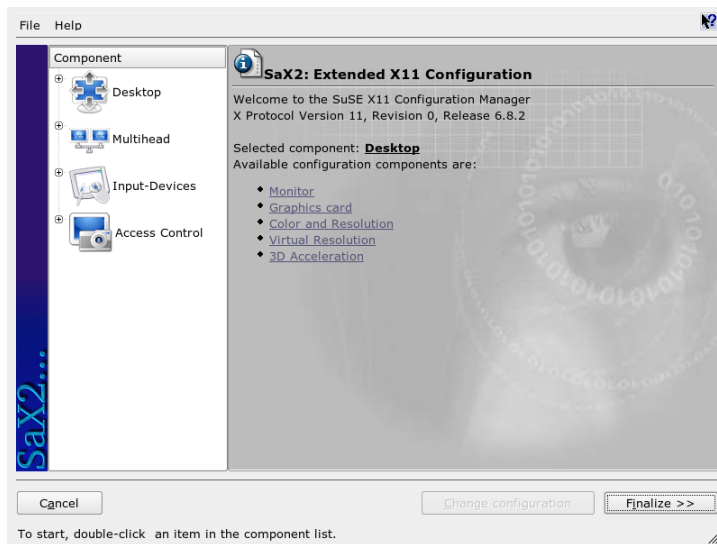


Figure 11.1: SaX2 的主要視窗

左方的瀏覽列中有四個主要項目：‘桌面’、‘多重顯示器’、‘輸入設備’與‘AccessX’。顯示器、圖形介面卡、色彩深度、解析度與畫面的位置與大小，可以在‘桌面’下設定。鍵盤、滑鼠、觸碰式螢幕顯示器與圖形板，可以在‘輸入設備’下設定。使用‘多重顯示器’可以設定多個畫面(請參閱節11.1.7, “多重顯示器”)。‘AccessX’是利用數字鍵盤，控制滑鼠指示點的實用工具。

選擇顯示器與圖形介面卡。通常，系統可自動偵測到顯示器與圖形介面卡。如果

未偵測到顯示器，就會自動進入顯示器選擇對話方塊。從許多廠商與設備的清單選擇您的顯示器，或是手動輸入顯示器手冊中指定的顯示器值。或是，選擇其中一個預先設定的VESA 模式。

完成顯示器與圖形介面卡的設定之後按一下‘完成’，然後測試您的設定。這個動作可以確保該設定是否適用於您的設備。如果影像不穩定，按下(ESC)立即終止測試，然後降低更新速率或色彩深度。無論您是否執行測試，只有在重新啓動X 伺服器之後才會啓用所有修改。

11.1.1 桌面

使用‘編輯組態’→‘內容’，就會出現有‘顯示器’、‘頻率’與‘進階’索引標籤的視窗。

‘顯示器’視窗左側可選擇廠商。視窗右側可選擇型號。如果磁片上有顯示器的Linux 驅動程式，按一下‘製造商磁片’即可安裝。

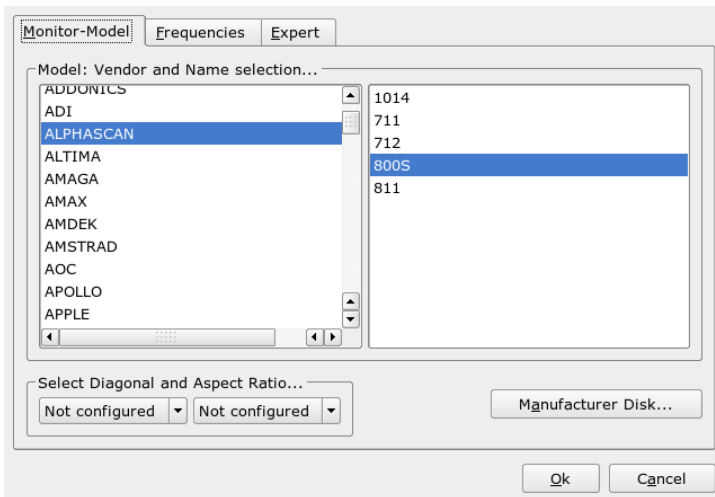


Figure 11.2: 顯示器選擇

‘頻率’在此輸入螢幕的水平與垂直頻率。垂直頻率是影像更新速率的另一種指定方式。一般可從型號取得可接受的值範圍，將它輸入於此。通常無需變更。

‘進階’ 此處可輸入螢幕的部分設定。在上方的選擇欄位中，可定義要用以計算螢幕解析度與螢幕幾何的方法。除非顯示器設定不正確而且顯示結果不穩定，否則請不要變更任何設定。此外，您可以變更顯示影像的大小和啓動省電模式DPMS。



設定顯示器頻率

雖然有安全機制，但手動變更允許的頻率時，還是應該小心謹慎。不正確的值可能會損壞您的顯示器。如果無法肯定，請參閱顯示器手冊。



11.1.2 圖形介面卡

圖形介面卡對話方塊有兩個索引標籤：‘一般’與‘進階’。圖形介面卡的廠商可在‘一般’的左側選擇，而型號則是在右側。

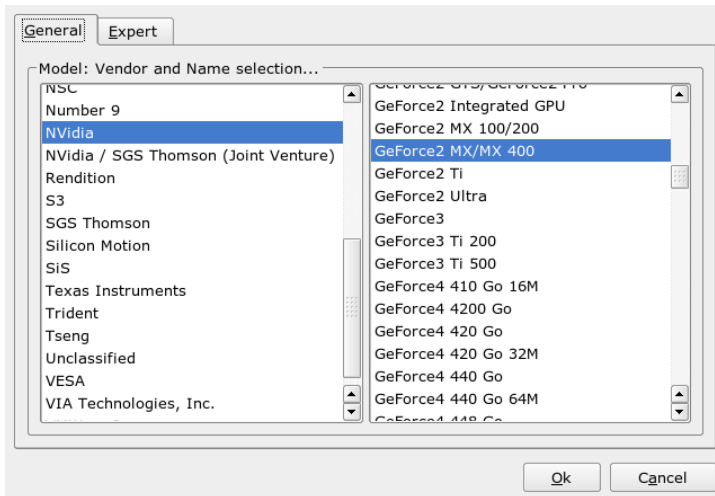


Figure 11.3: 選擇圖形介面卡

‘進階’提供其他可能的進階組態。右側有將螢幕向左轉或轉至垂直位置的選項(通常用於某些可轉向的TFT螢幕)。只有在操作多個螢幕時，才可以使用BusID項目。一般情況下，無需變更此處的任何設定。除非您曾使用過此欄位，而且知道各個選項的功能，否則不應該修改介面卡選項。如有需要，請參閱圖形介面卡文件。

11.1.3 色彩與解析度

這裡有‘色彩’、‘解析度’與‘進階’三個索引標籤可用。

‘色彩’根據使用的硬體而定，色彩深度的選項有16、256、32768、65536或16.7百萬色(4、8、15、16或24位元)。至少設為256色才會有適當的顯示品質。

‘解析度’這個模組提供硬體可正確顯示解析度與色彩深度的所有組合。在SUSE LINUX中，這可以降低因為不正確設定而損壞硬體的威脅。如果要手動變更解析度，請參閱硬體文件確定能夠顯示該值。

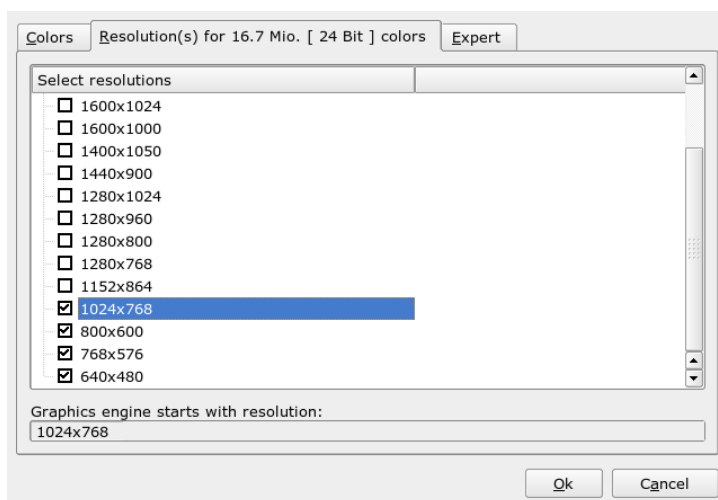


Figure 11.4: 設定解析度

‘進階’除了前一個索引標籤提供的解析度之外，這個索引標籤可讓您新增自己的解析度，隨後納入索引標籤的選項。

11.1.4 虛擬解析度

每個桌面都會有個特定的解析度能以顯示器的全螢幕大小顯示。此外，所設定的解析度也可能會超過螢幕的可見區域。如果滑鼠移動到桌面的邊界之外，虛擬的桌面就會顯示在螢幕上。這樣可以增加可用的工作空間。

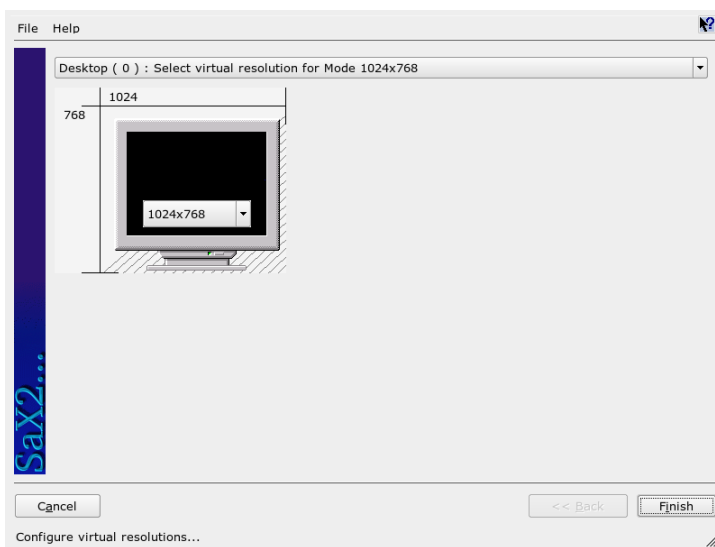


Figure 11.5: 設定虛擬解析度

虛擬解析度有兩種設定方法。若要使用‘拖放’來設定，請將滑鼠指示點移動到顯示器影像上，滑鼠指示點會變為十字。按住滑鼠左鍵並移動滑鼠，以放大與垂直解析度一致的光柵影像。如果您不確定桌面上要有多少垂直空間，最好使用這種方法。

就‘從快顯功能表選擇’而言，光柵影像中間的快顯功能表，會顯示目前使用的虛擬解析度。若要使用其中一種預設的虛擬解析度，從功能表中選擇即可。

11.1.5 3D 加速功能

如果未在啓始安裝期間，或在安裝新的圖形介面卡時啓動3D 加速功能，您可在此執行這項動作。

11.1.6 影像位置與大小

在這兩個索引標籤下方，可以使用箭頭精確調整影像大小與位置。請參閱圖形11.6, “調整影像幾何”。如果有多重顯示器環境(有一個以上的螢幕)，請使用‘下一個螢幕’切換到其他顯示器來調整大小與位置。按‘儲存’可儲存您的設定。

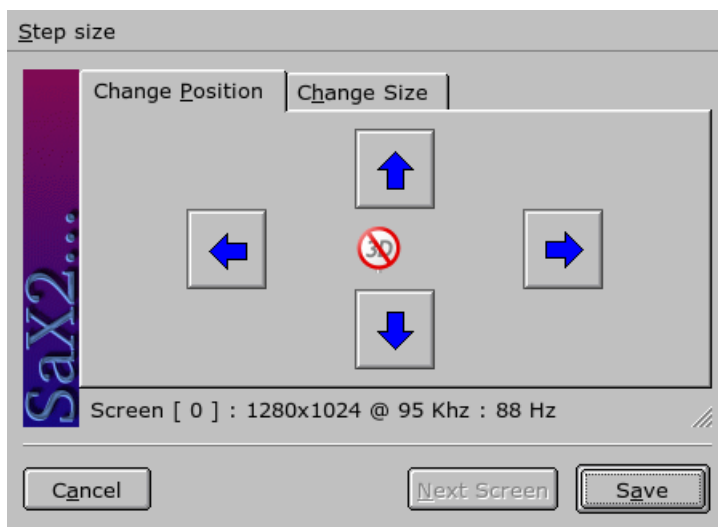


Figure 11.6: 調整影像幾何

11.1.7 多重顯示器

如果已在電腦中安裝多個圖形介面卡，或是具多重輸出功能的圖形介面卡，您的系統就可以連接一台以上的螢幕。如果是操作兩台螢幕，就稱為雙顯示器。兩台以上則稱為多重顯示器。SaX2 會自動偵測系統中的多個圖形介面卡，然後據以準備組態。在多重顯示器對話方塊中，設定多重顯示器模式和畫面安排。共提供三種模式：‘傳統’ (預設)、‘單一螢幕(Xinerama)’ 與‘複製模式’。

傳統多重顯示器 每台顯示器都代表個別的單位。可使用滑鼠指示點在螢幕間切換。

複製多重顯示器 在這個模式中，所有顯示器都會顯示相同的內容。只有在主要螢幕上才看得到滑鼠。

Xinerama 多重顯示器 所有螢幕會組合成單一的大畫面。程式視窗可以隨意放在所有螢幕上，或是縮放大小讓視窗填滿一台以上的顯示器。

多重顯示器環境的配置，可描述個別螢幕的排列與之間的關係。根據預設，SaX2 會遵循偵測到的圖形介面卡順序設定標準配置，由左自右安排所有螢幕。在多重顯示器工具的‘配置’對話方塊中，可以使用滑鼠在格線中移動螢幕符號，決定顯示器的安排方式。完成配置對話方塊之後，按一下‘測試’可以確認新的組態。

Linux 目前不支援 Xinerama 多重顯示器的 3D 環境。在此狀況下，SaX2 會關閉 3D 支援。

11.1.8 輸入設備

滑鼠 如果自動偵測失敗，可使用這個對話方塊手動設定您的滑鼠。如需型號的說明，請參閱您的滑鼠文件。從支援的滑鼠類型清單選擇您的型號，然後在數字鍵上按 **⑤** 來確認。

鍵盤 使用此對話方塊上方的選擇欄位，指定要使用何種鍵盤。然後選擇鍵盤配置的語言（語言特定的按鍵位置）。使用測試欄位來檢查特殊字元是否能正確顯示。

用來啟動和關閉輸入重音字元的核取方塊狀態，視個別語言而定，而且無需變更。按一下‘完成’可將新設定套用至系統。

觸碰式螢幕 目前 X.Org 只支援 Microtouch 與 Elo TouchSystems 觸碰式螢幕。SaX2 只能自動偵測到顯示器，而無法偵測到觸碰式設備。觸碰式設備視為一種輸入設備。

若要設定觸碰式設備，請啟動 SaX2，然後選擇‘輸入設備’→‘觸碰式螢幕’。按一下‘新增’，然後新增觸碰式螢幕。按一下‘完成’來儲存組態。您無需測試該組態。

觸碰式螢幕有多種不同選項，而且通常必須先進行校正。不幸地，Linux 沒有校正觸碰式螢幕的一般工具。標準組態包含適合觸碰式螢幕尺寸使用的預設值。通常不需要其他組態。

圖形板 目前 X.Org 支援的圖形板有限。SaX2 可以啓用連線到 USB 埠或序列埠的圖形板組態。從組態的觀點來看，圖形板只是一種類似滑鼠的輸入設備。

啓動SaX2，然後選擇‘輸入設備’→‘圖形板’。按一下‘新增’，從下列對話方塊選擇廠商，然後新增選擇清單中的圖形板。如果已經連接到筆或橡皮擦，就在核取方塊右邊標示記號。如果圖形板已連線到序列埠，請確認該埠。`/dev/ttyS0` 參照到第一序列埠。`/dev/ttyS1` 參照到第二序列埠。其他埠則使用類似的表示法。按一下‘完成’來儲存組態。

11.1.9 AccessX

若您的電腦未使用滑鼠，請啓動SaX2 和AccessX，以便使用數字鍵台來控制滑鼠指標。如需不同按鍵的功能說明，請參閱表格11.1, “AccessX — 使用數字鍵台操作滑鼠”。使用滑桿來設定按下按鍵時，滑鼠指示點移動的速度。

Table 11.1: AccessX — 使用數字鍵台操作滑鼠

金鑰	描述
⊖	選取滑鼠左鍵
⊗	選取滑鼠中鍵
⊖	選取滑鼠右鍵
⑤	在先前選取的滑鼠鍵呼叫按一下的事件。如果未選取其他滑鼠鍵，則會預先設定為滑鼠左鍵。事件發生之後，就會把該選擇預先設定為它的預設值。
+	除了按兩下事件，就像是⑤的動作
①	除了按住不動的事件外，就像是⑤的動作
Del	解除先前按①所呼叫的按住不動事件
⑦	將游標往左上移
⑧	將游標往上移
⑨	將游標往右上移
④	將游標往左移
⑥	將游標往右移
①	將游標往左下移
②	將游標往下移
③	將游標往右下移

11.1.10 若需詳細資訊

如需X Window 系統與其內容，請參閱章11, X Window System。

11.1.11 搖桿

從顯示的清單選擇製造商與型號，就可以使用此模組來設定您的搖桿。利用‘測試’來檢查您的搖桿是否能正確回應。測試對話方塊會顯示搖桿類比軸的三個圖表，並將四個標準按鈕標示出來。當您移動搖桿或按下按鈕時，應可在測試對話方塊中見到反應。因為搖桿通常會連接到音效卡，所以您也可以從音效卡組態來存取此模組。

11.1.12 鍵盤配置選擇

想要的鍵盤配置通常會與選擇的語言相符，但也可以不顧語言來選擇。請使用測試欄位來查看特殊字元是否能正確顯示，例如管狀符號|。

11.1.13 滑鼠

請使用此YaST 模組來設定您的滑鼠。選擇滑鼠的程序已經在安裝章節說明，請參閱節1.5.3, “滑鼠”。

11.2 X 組態最佳化

X.Org 是一個由「開放原始碼」(Open Source) 實作的X Window System。它是由X.Org Foundation 進一步發展而來，該組織還負責開發X Window System 的新技術與標準。

為了使可用的硬體，包括滑鼠、圖形卡、監視器以及鍵盤發揮最大效能，可以透過手動組態來達到最佳化。以下說明最佳化的一些層面。如需X Window System 組態的詳細資訊，請參閱目錄/usr/share/doc/packages/Xorg 中的各個檔案以及man xorg.conf。

┌

進行X Window System 組態時請務必小心。在完成組態之前，絕對不可以啓動X Window System。錯誤的組態會對硬體造成無法挽回的損害(固定頻率的監視器特別是如此)。本書的作者以及SUSE LINUX AG 無法對可能的損害負起任何責任。本資訊是仔細研究的結果，但不保證提及的所有方法都正確以及不會損害您的硬體。

└

SaX2 和xf86config 程式將建立xorg.conf 檔，預設的存放位置是/etc/X11。這是X Window System 的主要組態檔。請在此處找出和您的圖形卡、滑鼠以及監視器相關的所有設定。

下列段落描述組態檔/etc/X11/xorg.conf 的結構。每一個段落以關鍵字Section <指定> 開始，並以EndSection 結束。以下是重要段落的大綱。

xorg.conf 由多個段落組成，每一個段落處理組態的特定層面。段落的格式固定如下：

```
Section 指定
  項目 1
  項目 2
  項目 n
EndSection
```

提供的段落類型列出在表格11.2, “/etc/X11/xorg.conf 中的段落” 中。

Table 11.2: /etc/X11/xorg.conf 中的段落

類型	意義
Files (檔案)	這個段落描述字型與RGB 顏色表的使用路徑。
ServerFlags (伺服器標誌)	一般切換設定於此。
InputDevice (輸入設備)	輸入設備，例如，鍵盤和特殊輸入設備(觸控板、搖桿等)都是在這個段落設定。這個段落含有Driver (磁碟機)的重要參數以及定義Protocol (通訊協定)與Device (設備)的選項。

Monitor (顯示器)	描述使用的顯示器。這個段落的個別元素是名稱(將在Screen (螢幕) 定義中討論)、bandwidth (頻寬) 以及同步頻率限制(HorizSync (水平同步) 和VertRefresh (垂直更新))。設定值以MHz、kHz 和Hz 提供。通常，伺服器會拒絕和監視器規格不對應的模式行。這樣可以防止不小心將太高的頻率送往顯示器。
Modes (模式)	特定螢幕解析度的modeline (模式行) 參數儲存於此。這些參數可以由SaX2 依據使用者給定的值計算，通常不需要變更。如果您要連接固定頻率監視器，現在可以手動變更。請在HOWTO 檔/usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz 中找出每一個個別數值的意義之詳細資料。
Device (設備)	這個段落定義特定圖形卡。它是以本身描述名稱敘述。
Screen (螢幕)	這個段落和Monitor (顯示器) 以及Device (設備) 一起構成X.Org 所有必要的設定。在Display (顯示) 子段落，請指定虛擬螢幕的大小(Virtual (虛擬螢幕))、ViewPort (檢視埠)，以及該螢幕使用的Modes (模式)。
ServerLayout (伺服器配置)	這個段落定義單一或多重顯示的組態。這個段落連結輸入裝置InputDevice (輸入裝置) 和顯示設備Screen (螢幕)。

Monitor (顯示器)、Device (設備) 和Screen (螢幕) 將在以下詳細說明。如需其他段落的進一步資訊，請參閱X.Org 和xorg.conf 的手冊頁。

xorg.conf 中可能有多個不同的Monitor (顯示器) 和Device (設備) 段落。也可能有多個Screen (螢幕) 段落。下列ServerLayout (伺服器配置) 段落決定使用哪一個。

11.2.1 螢幕段落

首先，請仔細觀察螢幕段落，它由一個顯示器以及一個設備段落組成，以決定要使用的解析度與色彩深度。螢幕段落可能如同範例11.1, "/etc/X11/xorg.conf 檔的螢幕段落" 一般。

Example 11.1: /etc/X11/xorg.conf 檔的螢幕段落

```
Section "Screen"
    DefaultDepth 16
    SubSection "Display"
        Depth 16
        Modes "1152x864" "1024x768" "800x600"
        Virtual 1152x864
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "1280x1024"
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "640x480"
    EndSubSection
    SubSection "Display"
        Depth 8
        Modes "1280x1024"
    EndSubSection
    Device "Device[0]"
    Identifier "Screen[0]"
    Monitor "Monitor[0]"
EndSection
```

Identifier (識別碼) (此處為Screen[0]) 行對這個段落提供一個定義的名稱，以便對下列ServerLayout (伺服器配置) 段落提供唯一性參照。Device (設備) 和Monitor (顯示器) 行指定屬於這個定義的圖形卡和顯示器。它們透過本身對應的名稱或識別碼和Device (設備) 以及Monitor (顯示器) 段落連結。以下詳細討論這些段落。

如果伺服器未以特定的色彩深度啟動的話，請使用DefaultDepth (預設深度) 設定來選取伺服器使用的色彩深度。每一個色彩深度有一個Display (顯示器) 子段落。關鍵字Depth (深度) 指定這個子段落的有效色彩深度。Depth (深度) 的可用值有8、15、16和24。不是所有的X伺服器模組都支援所有這些值。

色彩深度之後，是一組Modes (模式) 段落中設定的解析度清單。X伺服器將由左而右檢查這個清單。對於每一個解析度，X伺服器會在Modes (模式) 段落搜尋適合的Modeline (模式行)。Modeline (模式行) 同時由顯示器與圖形卡的功能決定。Monitor (顯示器) 設定值將決定Modeline (模式行) 的結果。

第一個找到的解析度是Default mode (預設模式)。按Ctrl-Alt-+ (數字鍵盤) 可以切換到清單中右邊的下一個解析度。按Ctrl-Alt- (數字鍵盤) 可以向左切換。您可以在X執行中變更解析度。

Display (顯示器) 子段落最後一行的Depth 16 (深度16) 指虛擬螢幕的大小。虛擬螢幕可能的最大大小由圖形卡上安裝的記憶體數量以及所要的色彩深度

決定，而不是顯示器的最大解析度。因為目前的圖形卡都附有大量的視訊記憶體，您可以建立非常大的虛擬桌面。不過，如果將大部份的視訊記憶體用於虛擬桌面，將可能無法使用3D功能。例如，如果卡上有16 MB 視訊RAM，虛擬螢幕可用佔用4096x4096 像素，8 位元色彩深度。不過，特別是加速卡，不建議將所有記憶體用於虛擬螢幕，因為卡上的這個記憶體還要用於各種字型與圖形的快取處理。

11.2.2 設備段落

設備段落描述特定的圖形卡。xorg.conf 中的設備項目數沒有限制，但每一個設備的名稱要使用關鍵字Identifier (識別碼)來區別。有一個適用規則—如果您安裝多張圖形卡—段落將只依序編號。第一個稱為Device[0] (設備[0])，第二個稱為Device[1] (設備[1])，以下類推。以下檔案顯示一個使用Matrox Millennium PCI 圖形卡的電腦，其Device (設備)段落的例外情況：

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"
    Driver         "mga"
    Identifier     "Device[0]"
    VendorName    "Matrox"
    Option        "sw_cursor"
EndSection
```

如果您使用SaX2 進行組態，設備段落看起來應該和上述範例類似。Driver (驅動程式)和BusID (匯流排ID)都由電腦中安裝的硬體決定，並由SaX2 自動偵測。BusID (匯流排ID)定義安裝圖形卡的PCI或AGP 插槽。這個值和指令lspci 顯示的ID 相符。X 伺服器需要十進位格式的詳細資料，但lspci 以十六進位格式顯示這些值。

您可以使用Driver (驅動程式)參數來指定這個圖形卡使用的驅動程式。如果是Matrox Millennium 圖形卡，驅動程式模組稱為mga。X 伺服器會在定義於drivers 子目錄的Files 段落中的ModulePath 中搜尋。在標準安裝中，這是目錄/usr/X11R6/lib/modules/drivers。名稱會附加_drv.o，因此，如果是mga 驅動程式，將載入驅動程式檔案mga_drv.o。

X 伺服器或驅動程式的行為可以透過附加的選項來操作。設備段落中的選項sw_cursor 便是一個範例。它可以關閉硬體滑鼠游標並描述使用軟體的滑鼠指標。視驅動程式模組而定，各有不同的各種選項，可以在/usr/X11R6/lib/X11/doc 目錄內驅種程式模組的描述檔案中找到它們。通常，也可以在手冊頁找到有效的選項(man xorg.conf 和man X.Org)。

11.2.3 顯示器和模式段落

和Device (設備) 段落類似, Monitor (顯示器) 和Modes (模式) 段落分別描述一個顯示器。組態檔/etc/X11/xorg.conf 可以含有沒有個數限制的Monitor (顯示器) 段落。伺服器配置段落指定相關的Monitor (顯示器) 段落。

顯示器定義應該由有經驗的使用者進行設定。模式行是Monitor (顯示器) 段落的重要組成部份。模式行可設定對應解析度的水平與垂直計時。顯示器的內容, 特別是容許頻率, 儲存在Monitor (顯示器) 段落中。

┌

除非您對顯示器與圖形卡功能有深度的瞭解, 否則請不要變更模式行, 以免造成顯示器嚴重損害。

└

想自行開發顯示器描述的人員, 應該非常熟悉/usr/X11/lib/X11/doc 說明文件的內容。特別值得一提的是涵蓋視訊模式的段落。該段落詳細說明硬體功能以及如何建立模式行。

在今天, 需要手動指定模式行的的情況非常少見。如果您使用先進的多頻顯示器, 依規則, X 伺服器可以透過DDC 直接從顯示器讀取容許頻率以及最佳解析度值, 如SaX2 組態段落中的說明那樣。如果因某種原因而不適用, 請使用X 伺服器內附的一種VESA 模式。這樣幾乎所有的圖形卡與顯示器組合都可以運作。

11.3 安裝與設定字型

SUSE LINUX 上安裝其他字型非常簡單。只要將字型複製到位於X11 字型路徑中的任何目錄即可(請參閱節11.3.2, "X11 核心字型")。要以新的xft 字型處理系統使用字型, 它的安裝目錄必須是/etc/fonts/fonts.conf 中設定的目錄的子目錄(請參閱節11.3.1, "Xft")。

字型檔案必須手動方式(以root 身份) 複製到適當的目錄, 例如/usr/X11R6/lib/X11/fonts/truetype。此外, 這個作業可以透過「KDE 控制中心」中的KDE 字型安裝程式來執行。結果完全相同。

您也可以建立符號連結, 來替代複製實際字型。例如, 如果您擁有裝載Windows 分割區上的授權字型並且想要使用它們, 便可能需要這樣做。然後, 請執行SuSEconfig --module fonts。

SuSEconfig --module fonts 會執行程序檔 /usr/sbin/fonts-config，來處理字型的組態。如需這個程序檔的執行內容，請參閱程序檔的手冊頁(man fonts-config)。

點陣字型、TrueType 與OpenType 字型以及Type1 (PostScript) 字型的程序完全相同。所有這些字型類型可以安裝至任何目錄。只有CID-keyed 字型的程序稍有不同。相關資訊請參閱節11.3.3, "CID-Keyed 字型"。

X.Org 含有兩種完全不同的字型系統：舊式X11 核心字型系統和新設計的Xft 與fontconfig 系統。以下數節簡短描述這兩種系統。

11.3.1 Xft

從一開始，Xft 的程式設計人員便保證，會支援平滑美觀的可縮放字型。如果使用Xft，字型將由使用字型的應用程式處理，而不是如X11 核心字型系統那樣由X 伺服器處理。在這種方式，個別的應用程式需要存取實際字型檔並完全控制文字符號的處理方式。多種語言文字的精確顯示基礎便是由此構成。直接存取字型檔對於在列印中內嵌字型，來保證列印出的外觀和螢幕輸出一致非常有用。

在SUSE LINUX，KDE 和GNOME 二個桌面環境、Mozilla 以及其他許多應用程式已經預設使用Xft。越來越多的應用程式已經從舊X11 核心字型系統改用Xft。

Xft 使用fontconfig 程式庫來尋找字型以及操作字型的處理方式。fontconfig 的內容是由全域組態檔/etc/fonts/fonts.conf 和使用者專屬組態檔 ~/.fonts.conf 控制。這些fontconfig 組態檔每一個都必須以下列開頭

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

並以下列結束

```
</fontconfig>
```

要新增字型搜尋目錄，請附加下列行：

```
<dir>/usr/local/share/fonts/</dir>
```

不過，通常並不需要這樣做。依預設，使用者專屬目錄 ~/.fonts 已經輸入到/etc/fonts/fonts.conf 中。因此，安裝額外的字型時，只要將它們複製到 ~/.fonts 即可。

您還可以插入操作字型外觀的規則。例如，輸入


```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>>false</bool>
  </edit>
</match>
```

可以關閉所有字型的消除鋸齒效果，

```
<match target="font">
  <test name="family">
    <string>Luxi Mono</string>
    <string>Luxi Sans</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>>false</bool>
  </edit>
</match>
```

可以關閉特定字型的消除鋸齒效果。

依預設，大部份的應用程式都是使用字型名稱sans-serif (或相等的sans)、serif 或monospace。這些都不是真實的字型，而是可以解析為適當字型的別名，視語言設定而定。

使用者很容易在~/.fonts.conf 新增規則來將這些別名解析為屬意的字型：

```
<alias>
  <family>sans-serif</family>
  <prefer>
    <family>FreeSans</family>
  </prefer>
</alias>
<alias>
  <family>serif</family>
  <prefer>
    <family>FreeSerif</family>
  </prefer>
</alias>
<alias>
  <family>monospace</family>
  <prefer>
    <family>FreeMono</family>
  </prefer>
</alias>
```

因為幾乎所有的應用程式都預設使用這些別名，所以幾乎整個系統會受到影響。這樣，您幾乎可以簡單地隨處使用喜歡的字型而不必針對個別應用程式修改字型設定。

請使用指令 `fc-list` 來找出已安裝且可用的字型。例如，指令 `fc-list ""` 可傳回所有字型的清單。要找出含有希伯來文(`:lang=he`)所有文字符號的可縮放字型(`:outline=true`)、這些字型的名稱(`family`)、樣式(`style`)、粗細(`weight`)以及包含這些字型的檔案名稱，請輸入下列指令：

```
fc-list ":lang=he:outline=true" family style weight file
```

這個指令的輸出看起來如下：

```
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBold.ttf: FreeSans:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBoldOblique.ttf: FreeMono:style=BoldOblique: weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSerif.ttf: FreeSerif:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBoldItalic.ttf: FreeSerif:style=BoldItalic: weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansOblique.ttf: FreeSans:style=Oblique:weight=8 0
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifItalic.ttf: FreeSerif:style=Italic:weight=8 0
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoOblique.ttf: FreeMono:style=Oblique:weight=8 0
/usr/X11R6/lib/X11/fonts/truetype/FreeMono.ttf: FreeMono:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSans.ttf: FreeSans:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBold.ttf: FreeSerif:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBoldOblique.ttf: FreeSans:style=BoldOblique: weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBold.ttf: FreeMono:style=Bold:weight=200
```

可以使用 `fc-list` 來查詢的重要參數：

Table 11.3: fc-list 的參數

參數	意義及可用值
<code>family</code>	(系列字型系列的名稱，例如，FreeSans。
<code>foundry</code>	(製造廠的製造商，例如，urw。
<code>style</code>	(樣式) 字型樣式，例如，Medium (中)、Regular (一般)、Bold (粗體)、Italic (斜體)、Heavy (濃)。
<code>lang</code>	(語言) 字型支援的語言，例如，de 表示德文、ja 表示日文、zh-TW 表示繁體中文、zh-CN 表示簡體中文。
<code>weight</code>	(粗細) 字型粗細，例如，80 表示一般，200 表示粗體。
<code>slant</code>	(斜度) 通常，斜度 0 表示沒有斜度，100 表示斜體。
<code>file</code>	(檔案) 字型檔案的名稱。
<code>outline</code>	(外框) <code>true</code> (真) 表示外框字型， <code>false</code> (偽) 表示其他字型。
<code>scalable</code>	(縮放) <code>true</code> (真) 表示縮放字型， <code>false</code> (偽) 表示其他字型。
<code>bitmap</code>	(點陣) <code>true</code> (真) 表示點陣字型， <code>false</code> (偽) 表示其他字型。
<code>pixelsize</code>	(點數) 表示的字型大小。這個選項和 <code>fc-list</code> 一起使用時只對點陣字型有意義。

11.3.2 X11 核心字型

目前，X11 核心字型系統不僅支援點陣字型，也支援縮放字型如Type1 字型、TrueType 與OpenType 字型，以及CID-keyed 字型。Unicode 字型的支援已經有一段時間。X11 核心字型系統最早於1987 年針對X11R1 的黑白點陣字型處理而開發。以上所述的擴充是在之後新增的。

縮放字型僅只支援不需要消除鋸齒及子像素(subpixel) 處理的字型，含有更多語言的文字符號的縮放字型所需的載入時間也越長。Unicode 字型的處理速度較慢而且需要更多的記憶體。

X11 核心字型系統具有某些先天性的弱點。它已經過時，而且無法以現有方法擴充。保留這種字型是爲了回溯相容，但最好儘可能使用更先進的Xft 和fontconfig 系統。

爲了能夠進行作業，X 伺服器必須知道它有哪些字型可用，以及系統中的哪些地方可以找到這些字型。這是由FontPath 變數負責處理，該變數包含所有有效系統字型目錄的路徑。每個目錄中的fonts.dir 檔案，可以列出目錄中可用的字型有哪些。FontPath 是X 伺服器在啓動時產生的。它會在/etc/X11/xorg.conf 組態檔案的FontPath 項目中，搜尋一個有效的fonts.dir 檔案。這些項目會在Files 區段中找到。使用xset q 顯示實際的FontPath。也可以使用xset 在執行時變更此路徑。若要新增其他路徑，請使用xset +fp <path>。若要移除不要的路徑，請使用xset -fp <path>。

如果X 伺服器已經啓動，裝載目錄中新安裝的字型可以透過指令xset fp rehash 來設成可用。這個指令以SuSEconfig --module fonts 執行。由於指令xset 必須存取執行中的X 伺服器，因此只有在SuSEconfig --module fonts 是從可存取之執行中X 伺服器的外圍程序啓動時才有效。最簡單的達成方法是輸入su 與root 密碼來取得root 權限。su 可以將啓動X 伺服器的使用者的存取權限轉移給root 外圍程序。要檢查字型是否安裝正確以及是否可以透過X11 核心字型系統來使用，請使用xlsfonts 指令來列出所有可用字型。

依預設，SUSE LINUX 使用UTF-8 語言環境。因此，最好使用Unicode 字型(在xlsfonts 的輸出中，字型名稱的結尾爲iso10646-1)。xlsfonts | grep iso10646-1 可以列出所有可用的Unicode 字型。SUSE LINUX 附隨的Unicode 字型絕大部份都具有歐洲語言(舊編號方式爲iso-8859-*) 所需的文字符號。

11.3.3 CID-Keyed 字型

和其他字型類型相反，您無法將CID-keyed 字型安裝在任一目錄。CID-keyed 字型必須安裝在/usr/share/ghostscript/Resource/CIDFont。這和Xft 與fontconfig 無關，但是對Ghostscript 和X11 核心字型系統卻是必要的。

請參閱<http://www.xfree86.org/current/fonts.html> 來取得X11的字型詳細資訊。

11.4 OpenGL — 3D 組態

11.4.1 硬體支援

SUSE LINUX 包括數個3D 硬體支援程式。表格11.4, “支援的3D 硬體” 提供概觀。

Table 11.4: 支援的3D 硬體

OpenGL 驅動程式	支援的硬體
nVidia	nVidia 晶片：除了Riva 128(ZX) 以外的所有晶片
DRI	3Dfx Voodoo Banshee , 3Dfx Voodoo-3/4/5 , Intel i810/i815/i830M , Intel 845G/852GM/855GM/865G,915, Matrox G200/G400/G450/G550 , ATI Rage 128(Pro)/Radeon (高達9250)

如果您第一次安裝YaST，而且YaST 偵測到3D 支援，3D 加速功能就可以在安裝時啓動。要使用nVidia 繪圖晶片，必須先安裝nVidia 驅動程式。要這樣做，請在YOU (YaST Online Update) 中選取nVidia 驅動程式。因為授權限制，nVidia 驅動程式未包含在配送中。

如果執行的是更新而不是全新安裝，或是需要設定3Dfx 附加圖形介面卡(Voodoo Graphics 或Voodoo-2)，那麼設定3D 硬體支援的程序是不一樣的。這要視使用的OpenGL 驅動程式而定。詳細資訊請見下一章節：

11.4.2 OpenGL 驅動程式

OpenGL 驅動程式nVidia 和DRI 可以透過SaX2 輕輕鬆鬆設定。要使用nVidia 介面卡，必須先安裝nVidia 驅動程式。輸入指令3Ddiag 以確認nVidia 或DRI 組態是否正確。

基於安全因素，只有屬於video 群組的使用者才有權存取3D 硬體。因此，請確定所有本地使用者皆是此群組的成員。否則，OpenGL 應用程式會使用速度緩慢的軟體轉譯後援OpenGL 驅動程式。使用指令id 檢查目前使用者是否屬於video 群組。如果不是，請用YaST 將使用者新增至群組。

11.4.3 診斷工具3Ddiag

您可以使用診斷工具3Ddiag 來驗證SUSE LINUX 的3D 組態。這個指令行工具必須使用終端機啓動。輸入3Ddiag -h 以列出3Ddiag 的可能選項。

要檢驗X.Org 組態，工具會檢查是否已安裝3D 支援所需要的套件，以及檢查是否使用正確的OpenGL 程式庫和GLX 擴充。如果您收到錯誤訊息，請依照診斷工具3Ddiag 的指示進行操作。如果一切無誤，您只會在畫面上看到完成訊息。

11.4.4 OpenGL 測試公用程式

要測試OpenGL，程式glxgears 以及tuxracer 和armagetron 等遊戲(相同名稱的套件) 十分有用。如果已啓動3D 支援，在很新的電腦上應該可以玩的很順暢。沒有3D 支援，這些遊戲執行速度會相當慢(投影片效果)。請使用glxinfo 指令來檢驗3D 是否在作用中，如果在作用中，就會輸出一行字，其中包括direct rendering: YES。

11.4.5 疑難排解

如果OpenGL 3D 測試結果為否定(遊戲無法流暢的運作)，請使用診斷工具3Ddiag 確認組態沒有錯誤存在(錯誤訊息)。如果更正這些錯誤仍然沒有用，或者一直沒有顯示失敗訊息，請查閱X.Org 日誌檔。

您往往會在X.Org 檔案/var/log/Xorg.0.log 中發現DRI 已停用這一行字。確實原因必須仔細檢查日誌檔才能找到，而且這項工作需要一些經驗才行。

在類似情況下，不會有任何組態錯誤，因為3Ddiag 已經偵測過了。因此在此情況下，唯一選擇是使用DRI 驅動程式的軟體轉譯後援軟體，該軟體不提供3D 硬體支援功能。如果發生OpenGL 表示錯誤或不穩定，您可以不用3D 支援逕行操作。使用SaX2，完全停用3D 支援。

11.4.6 安裝支援

除了DRI 驅動程式的軟體轉譯後援外，Linux 的所有OpenGL 驅動程式仍在開發階段，因而尚屬實驗性質。因為對Linux 的3D 硬體加速需求量很高，所以配送中包括驅動程式。由於OpenGL 驅動程式的實驗性質，SUSE 無法為3D 硬體加速的設定提供任何安裝支援，也無法對任何相關問題提供進一步的協助。圖形化使用者介面(X Window 系統) 的基本組態不包括3D 硬體加速組態。如果您遇到3D 硬體加速問題，建議完全停用3D 支援。

11.4.7 其他線上文件

如需關於DRI 的資訊，請參閱/usr/X11R6/lib/X11/doc/README.DRI (xorg-x11-doc)。可在<http://ftp.suse.com/pub/suse/i386/supplementary/X/nvidia-installer-HOWTO.html> 找到nvidia 驅動程式安裝相關資訊。

印表機操作

本章節提供關於操作印表機的一般資訊，並協助找出在網路中操作印表機的適當解決方案。此外，並特別強調CUPS作業。其中包含詳細的疑難排解小節，簡述印表機操作中最常見的問題，並描述避免這些問題的方法。

12.1	準備與其他考量	206
12.2	列印系統的工作流程	207
12.3	連線印表機的方法和通訊協定	207
12.4	安裝軟體	208
12.5	設定印表機	208
12.6	應用程式的組態	213
12.7	SUSE LINUX 中的特殊功能	214
12.8	疑難排解	218

12.1 準備與其他考量

CUPS 是 SUSE LINUX 中的標準列印系統。CUPS 為高度使用者導向。在許多情況下，可與 LPRng 相容，或稍做努力便可與其搭配。基於相容性的原因，LPRng 包括在 SUSE LINUX 中。

印表機可藉由介面(例如 USB 或網路)和印表機語言加以區分。在購買印表機時，請確認硬體和適當的印表機語言可支援印表機的介面。印表機可根據下列三種印表機語言來分類：

PostScript 印表機 Linux 和 Unix 內部列印系統以 PostScript 印表機語言產生和處理大部分列印工作。此語言已經很舊且非常有效率。如果印表機可直接處理 PostScript 文件，且不需轉換到列印系統中其他階段，潛在錯誤來源的次數便會減少。因為 PostScript 受限於高額的授權成本，這些印表機通常較無 PostScript 解譯器的印表機昂貴。

標準印表機(PCL 和 ESC/P 語言) 雖然這些印表機語言很舊，它們仍在擴充以處理印表機的新功能。如果已知印表機語言，列印系統可以藉由 Ghostscript 的協助，將 PostScript 工作轉換為對應的印表機語言。此處理階段稱為解譯。最知名的語言為 PCL，大多為 HP 印表機與其複製品所使用；另一個是 ESC/P，大多為 Epson 印表機所使用。Linux 通常支援這些印表機語言，並可產生不錯的列印效果。Linux 可能無法處理非常新且新潮的印表機的部分功能，因為「開放原始碼」開發人員仍在研究這些功能。除了 HP 所開發的 `hpijs` 驅動程式之外，目前沒有開發 Linux 驅動程式並提供 Linux 散發者使用的印表機製造商得到「開放原始碼」授權。這些印表機大部分屬於中等價位。

專用印表機(通常為 GDI 印表機) 專用印表機通常僅可使用一或數個 Windows 驅動程式。這些印表機不支援任何常見印表機語言，它們所使用的印表機語言在發行新版的機型時會有變更。請參閱節 12.8.1, “沒有標準印表機語言模式支援的印表機” 取得更多資訊。

在購買新印表機之前，請參考下列來源以檢查您想要購買的印表機之支援性：

- <http://cdb.suse.de/>— SUSE LINUX 印表機資料庫
- <http://www.linuxprinting.org/>— LinuxPrinting.org 印表機資料庫
- <http://www.cs.wisc.edu/~ghost/>— Ghostscript 網頁
- `/usr/share/doc/packages/ghostscript/catalog.devices`— 包括的驅動程式

線上資料庫會永遠顯示最新的Linux支援狀態。但是，Linux版本僅可與生產期間可用的驅動程式整合。因此，目前被評比為“完全支援”的印表機，在最新的SUSE LINUX版本發行之後，可能就不在此狀態。因此，資料庫不一定能指出正確狀態，而僅提供估計值。

12.2 列印系統的工作流程

使用者會建立列印工作。列印工作的組成元素為要列印的資料加上暫存序列器的資訊(例如印表機的名稱或是印表機佇列的名稱)，以及非必要的過濾器資訊(例如，印表機的特定選項)。

每個印表機皆存在一個專用的印表機佇列。暫存序列器會在佇列中列印工作，直到所需的印表機已準備好接收資料。當印表機備妥時，暫存序列器會透過過濾器與後端，傳送資料至印表機。

過濾器會將使用者想要列印的資料(ASCII、PostScript、PDF、JPEG等)轉換成印表機特定的資料(PostScript、PCL、ESC/P等)。印表機的特性描述在PPD檔案中。PPD檔案含有印表機特定選項以及在印表機上啓用它們所需的參數。過濾器系統可確保啓用使用者所選取的選項。

如果您是使用PostScript印表機，過濾器系統會將資料轉換為印表機特定的PostScript。這並不需要印表機驅動程式。如果您是使用非PostScript印表機，過濾器系統會將資料轉換為使用Ghostscript的印表機特定資料。這將需要印表機適用的Ghostscript印表機驅動程式。後端會從過濾器接收印表機特定的資料，然後將它傳送至印表機。

12.3 連線印表機的方法和通訊協定

有各種方法可將印表機連線到系統。CUPS列印系統的組態無法辨識本地印表機和透過網路連線到系統的印表機。在Linux中，本地印表機必須依照印表機製造商的手冊所述方式連線。CUPS支援序列埠、USB、並列埠和SCSI連線。如需關於印表機連線的詳細資訊，請閱讀「支援資料庫」文章*CUPS in a Nutshell*，網址為<http://portal.suse.com>。可在搜尋對話方塊中輸入*cups*以找尋文章。

纜線連線到機器

在將印表機連線到機器時，請不要忘記只有USB設備可在操作中插上和拔除。在變更其他連線之前，系統必須關閉。

12.4 安裝軟體

PPD (PostScript 印表機描述) 為描述內容(如解析度)和選項(如雙面列印模組的可用性)的電腦語言。這些描述是使用CUPS中各種印表機選項所需。沒有PPD檔案，列印資料會被轉送給處於“raw”狀態的印表機，這通常不是想要的狀態。在SUSE LINUX安裝期間，會預先安裝許多PPD檔案，以便即使在沒有PostScript支援的情況下仍能使用印表機。

若要設定PostScript印表機，最好的方法是取得適當的PPD檔。在標準安裝範圍中自動安裝的套件manufacturer-PPDs提供許多PPD檔案。請參閱節12.7.4, “各種套件中的PPD檔案”與節12.8.2, “PostScript印表機沒有可用的PPD檔案”。

新的PPD檔案可儲存在目錄/usr/share/cups/model/中，或以YaST(請參閱節12.5.1, “手動組態”)新增到列印系統。之後，便可在安裝期間選取PPD檔案。

請小心，印表機製造商是否除了修改組態檔之外，要您安裝整個軟體套裝。首先，此種安裝會導致遺失SUSE LINUX所提供的支援，其次，印表機指令可能會以不同方式運作，使系統將無法處理其他製造商的設備。基於此原因，不建議安裝製造商軟體。

12.5 設定印表機

在將印表機連線到電腦並安裝軟體之後，請在系統中安裝印表機。這應該使用SUSE LINUX隨附的工具來執行。因為SUSE LINUX非常強調安全性，協力廠商工具通常難以應付安全性限制，結果產生的麻煩多過好處。

12.5.1 本地印表機

在您登入時，如果偵測到無法辨識的本地印表機，YaST會開始設定它。所用的對話方塊，與下列組態描述相同。

若要設定印表機，請選擇YaST 控制中心裡的‘硬體’→‘印表機’。這樣會開啓主要印表機的設定視窗，所偵測到的設備會列在視窗上方。下方會列出目前已設定的所有佇列。如果未偵測到您的印表機，請以手動方式設定。

如果‘印表機’項目在YaST 控制中心內沒有提供，則最有可能是未安裝yast2-printer 套件。若要解決此問題，請安裝yast2-printer 套件，然後重新啓動YaST。

自動組態

如果並列埠或USB 埠可以自動設定，並偵測到連線的印表機，YaST 就能夠自動設定印表機。印表機資料庫還必須包含硬體自動偵測時YaST 擷取的印表機ID 字串。如果硬體ID 與型號指定不同，請以手動方式選擇型號。

若要確保一切功能正常，每個組態都要用YaST 的列印測試功能檢測。YaST 的測試頁也提供受測組態的重要資訊。

手動組態

如果不符合自動組態的需求，或是您要自訂安裝，請以手動方式設定印表機。視自動偵測的成功程度和在資料庫中找到的印表機型號資訊多寡，YaST 可能可以決定正確的自動設定，或至少做出合理的預先選擇。

下列參數必須設定：

硬體連線(連接埠) 硬體連線的組態端視YaST 能否在硬體自動偵測時找到印表機。如果YaST 能夠自動偵測到印表機的型號，就可以假設印表機連線在硬體層次上工作正常，因此不需要變更這方面的設定。如果YaST 無法自動偵測到印表機的型號，硬體層次就可能有些連線的問題。在此情況下，需要以手動方式設定連線。

佇列名稱 發出列印指令時會用到佇列名稱。這個名稱應該十分簡短，並且只能包含小寫字母和數字。

印表機型號和PPD 檔案 所有印表機專用的參數，像是要使用的Ghostscript 驅動程式和驅動程式的印表機篩選器參數，都儲存在PPD (PostScript 印表機描述, PostScript Printer Description) 檔案中。若需更多PPD 檔案的詳細資訊，請參閱節12.4, “安裝軟體”。

很多印表機型號都有數種PPD檔案可用。例如，如果數種Ghostscript驅動程式都可搭配特定機種使用。您選擇一個製造商和一種型號時，YaST會選擇與該印表機對應的PPD檔案。如果該型號有數種PPD檔案可用，YaST會預設為其中一種(通常標示為建議)。您可以在選取‘編輯’後，變更預設的PPD檔案。

至於非PostScript的型號，所有印表機專用的資料都由Ghostscript驅動程式產生。因此，印表機組態是決定輸出品質最重要的一項因素。所選擇之Ghostscript驅動程式(PPD檔案)種類和為其指定的選項都會影響到列印成品。需要時可在選取‘編輯’後，變更其他選項(PPD檔案中所具備者)。

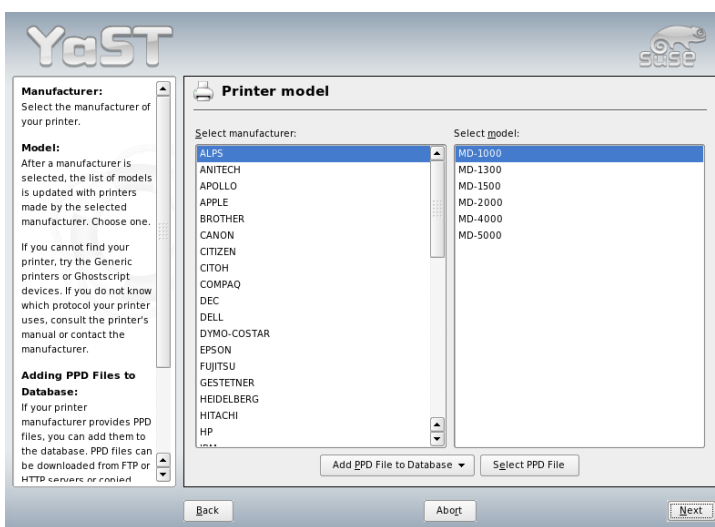


Figure 12.1: 選擇印表機型號

每次都要列印測試頁，來檢查您的設定是否如預期般有效。例如，如果輸出時出現問題，而且有幾頁幾乎空白，您應該移除所有紙張，然後停止YaST的測試，停止列印。

如果印表機資料庫不包括您的型號項目，您可以選取‘新增PPD檔案至資料庫’以新增新PPD檔案，或使用一般PPD檔案集合，讓印表機使用一種標準印表機語言來工作。若要這麼做，請選取‘未知的廠商’為您的印表機製造商。

進階設定 正常情況下，您不需要變更這些設定中的任何項目。

以指令行工具設定印表機

若要依照節12.5.3, “以指令行工具設定” 中所述以指令行工具手動設定印表機，您需要由後端(如USB)和參數(如/dev/usb/lp1)所組成的設備URI (Uniform Resource Locator，一致資源定址器)。例如，完整的URI可為parallel:/dev/lp0 (連線到第一並列埠的印表機) 或usb:/dev/usb/lp1 (第一個偵測到連線到USB埠的印表機)。

12.5.2 網路印表機

網路印表機可支援各種通訊協定，有些甚至可同時支援。雖然大部分支援的通訊協定為標準的，部分製造商會因為測試系統無法正確地執行標準，或因為想要提供標準無法提供的特定功能，而擴充(修改)標準。然後製造商僅對少數作業系統提供驅動程式，以減少那些系統的困難。不幸地，他們很少提供Linux 驅動程式。目前的情況是，您無法以每一個通訊協定均能在Linux 中順暢執行的假設來行事。因此，您必須試驗各種選項以達到功能性組態。

CUPS 支援socket、LPD、IPP 和smb 通訊協定。此處有一些關於這些通訊協定的詳細資訊：

socket *Socket* 指不須先執行資料信號交換，而將資料傳送到網際網路插槽的連線。經常使用的插槽連接埠號碼為9100 或35。設備URI 的範例為socket://host-printer:9100/。

LPD (行列式印表機精靈，Line Printer Daemon)

經過實驗的LPD 通訊協定描述於RFC 1179 中。在此通訊協定之下，部分工作相關資料(如印表機佇列的ID) 會在傳送實際列印資料之前傳送。因此，在設定資料傳輸的LPD 通訊協定時，必須指定印表機佇列。不同印表機製造商的執行具有足夠彈性接受任何名稱做為印表機佇列。如有需要，印表機手冊應該會指出要使用的名稱。通常使用LPT、LPT1、LP1或相似名稱。當然，LPD 佇列也可以在CUPS 系統中不同Linux 或Unix 主機上設定。LPD 服務的連接埠號碼為515。設備URI 的範例為lpd://host-printer/LPT1。

IPP (網際網路列印通訊協定，Internet printing protocol)

IPP 是相對較新的(1999) 通訊協定，以HTTP 通訊協定為基礎。有了IPP，可比使用其他通訊協定傳輸更多工作相關資料。CUPS 使用IPP 進行內部資料傳輸。這是在兩個CUPS 伺服器之間轉送佇列偏好的通訊協定。正確設定IPP 必須要有列印佇列的名稱。IPP 的連接埠號碼為631。設備URI 的範例為ipp://host-printer/ps 與ipp://host-cupsserver/printers/ps。

SMB (Windows 共用) CUPS 也支援在連接到Windows共用的印表機上列印。此用途使用的通訊協定為SMB。SMB使用連接埠號碼137、138和139。設備URI的範例為smb://user:password@workgroup/server/printer、smb://user:password@server/printer。

必須在設定組態之前決定印表機支援的通訊協定。如果製造商未提供所需資訊，可使用nmap指令(nmap套件)來猜測通訊協定。nmap會檢查主機上開啓的通訊埠。例如：

```
nmap -p 35,137-139,515,631,9100-10000 printerIP
```

12.5.3 組態工作

也可以使用YaST或指令行工具來執行組態工作。

使用YaST設定網路中的CUPS

網路印表機應該以YaST設定。YaST可幫助組態，而且是處理CUPS安全性限制的最佳配備(請參閱節12.7.2,“CUPS網路前端管理員”)。

如需在網路上安裝CUPS的說明，請參閱<http://portal.suse.com>上「支援資料庫」的文章*Nutshell*中的CUPS (*CUPS in a Nutshell*)。

以指令行工具設定

或者，也可以使用指令行工具(例如，lpadmin與lpoptions)設定CUPS。如果已經做好預備工作(如果您知道PPD檔案和設備名稱)，則必須採取下列步驟：

```
lpadmin -p queue -v device-URI -P PPD-file -E
```

請勿使用-E做為第一選項。對於所有CUPS指令，第一個引數-E設定使用加密連線。若要啓用印表機，必須依照下列範例所示使用-E：

```
lpadmin -p ps -v parallel:/dev/lp0 -P \  
/usr/share/cups/model/Postscript.ppd.gz -E
```

下列範例是設定網路印表機：

```
lpadmin -p ps -v socket://192.168.1.0:9100/ -P \  
/usr/share/cups/model/Postscript-levell.ppd.gz -E
```

修改選項

在系統安裝期間，某些選項會設成預設。可針對每一個列印工作修改這些選項(視所使用的列印工具而定)。也可以使用YaST 變更這些預設選項。使用指令行工具，可依下列方式設定預設選項：

1. 首先，列出所有選項：

```
lpoptions -p queue -l
```

範例：

```
解析度 / 輸出解析度：150dpi *300dpi 600dpi
```

前面的星號(*) 表示預設選項已經啓動。

2. 以lpadmin 變更選項：

```
lpadmin -p queue -o Resolution=600dpi
```

3. 檢查新設定：

```
lpoptions -p queue -l
```

```
解析度 / 輸出解析度：150dpi 300dpi *600dpi
```

12.6 應用程式的組態

應用程式與任何指令行工具一樣，都依賴現有印表機佇列。通常並不需要爲了特別的應用程式而重新設定印表機，因爲您應該可以使用既有的佇列，由應用程式進行列印。

12.6.1 由指令行開始列印

若要由指令行進行列印，請輸入指令 `lp -d <佇列名稱> <檔案名稱>`；請以相對應的名稱來取代 `<佇列名稱>` 及 `<檔案名稱>`。

12.6.2 使用指令行工具由應用程式列印

有些應用程式有賴 `lp` 指令來進行列印。在此情況下，請在應用程式列印對話方塊中(但通常並未指定〈檔案名稱〉)輸入正確的指令，例如 `lp -d 〈佇列名稱〉`。若要讓它與KDE 程式一起作業，請啓用‘由外部程式列印’。否則請勿輸入列印指令。

12.6.3 使用CUPS 列印系統

`xpp` 和KDE 程式 `kprinter` 等工具備有圖形介面，以便選擇佇列，並可設定PPD 檔案中可用的CUPS 標準選項和印表機專用的選項。您可以把 `kprinter` 當做非KDE 應用程式的標準列印介面，方法是在這些應用程式的列印對話方塊中指定 `kprinter` 或 `kprinter --stdin` 做為列印指令。應用程式自己的動作會決定選擇這兩個指令中的哪一個。如果設定正確，應用程式會在發出列印工作時呼叫 `kprinter` 對話方塊，您就可以使用對話方塊來選取佇列，並設定其他列印選項。但前提是該應用程式自有的列印設定與 `kprinter` 的列印設定不相衝突，而且列印選項啓用後只會經由 `kprinter` 做變更。

12.7 SUSE LINUX 中的特殊功能

CUPS 的許多功能已經可適用於SUSE LINUX。此處涵蓋部份最重要的變更。

12.7.1 CUPS 伺服器與防火牆

有數種方式可將CUPS 設為網路伺服器的用戶端。

- 對於網路伺服器上的每一個佇列，您可以設定一個本地佇列，透過它將所有工作轉送給對應網路伺服器。通常不建議使用此方法，因為只要網路伺服器的組態變更，就必須重新設定所有用戶端機器。
- 列印工作也可直接轉送給一台網路伺服器。對於此類型的組態，請勿執行本地的CUPS 精靈。`lp` 或其他程式的對應程式庫呼叫可將工作直接傳送給網路伺服器。但是，如果您也想在本地印表機列印，則此組態無法運作。
- CUPS 精靈可聆聽其他網路伺服器傳送來宣告可用佇列的IPP 廣播封包。這是透過遠端CUPS 伺服器列印的最佳CUPS 組態。但是，這有一個風險，攻擊者使用佇列傳送精靈IPP 廣播，而本地精靈存取到偽造的佇列。如果

這樣，則佇列會與本地伺服器上另一個佇列以相同名稱顯示，會較早收到IPP 封包，工作的擁有者會相信工作已傳送到本地伺服器，而事實上卻是傳送到攻擊者的伺服器。若要使用此方法，必須對內送封包開啓連接埠631/UDP。

YaST 可以掃描所有的網路主機以查看它們是否提供此服務，以及傾聽IPP 廣播，以便找到CUPS 伺服器。第二個方法也可在系統安裝期間用來尋找CUPS 伺服器以提供建議。它需要為內送封包開啓631/UDP 埠。

建議對話方塊中所顯示的防火牆預設值，是在任何介面上拒絕IPP 廣播。因此，第二個偵測遠端佇列的方法和第三個存取遠端佇列的方法均沒有作用。所以，必須將一個介面標示為internal (預設開啓連接埠) 或明確地開啓external 介面的連接埠，來修改防火牆組態。基於安全因素，預設不開啓任何連接埠。使用第二種方法開啓連接埠以設定遠端佇列的存取可能會造成安全上的風險，因為攻擊者有可能會廣播使用者可能接受的伺服器。

建議的防火牆組態必須修改為在安裝期間啓用CUPS 以偵測遠端佇列，以及在一般操作期間從本地系統存取不同的遠端伺服器。或者，使用者可主動掃描本地網路主機或手動設定所有佇列，以偵測CUPS 伺服器。但是，由於上述因素，不建議使用此方法。

12.7.2 CUPS 網路前端管理員

若要以網路前端(CUPS) 或印表機管理工具(KDE) 來管理，使用者root 必須以CUPS 管理群組sys 和CUPS 密碼設為CUPS 管理員。使用以下指令以root 身分執行：

```
lppasswd -g sys -a root
```

如果無法執行，則無法管理網路介面或管理工具，因為如果沒有設定CUPS 管理員，驗證便失敗。除了root 之外，也可以指定其他使用者為CUPS 管理員(請參閱節12.7.3, "CUPS 列印服務(cupsd) 中的變更")。

12.7.3 CUPS 列印服務(cupsd) 中的變更

這些變更一開始是針對SUSE LINUX 9.1 所套用。

cupsd 以使用者lp 執行

啓動時，cupsd 從使用者root 變更為使用者lp。如此可提供較高層級的安全性，因為CUPS 列印服務不會以未受限制的權限執行，而只會以列印服務所需的權限執行。

不過，無法透過/etc/shadow 執行驗證(密碼檢查)，因為lp 沒有/etc/shadow 的存取權。而必須使用透過/etc/cups/passwd.md5 的CUPS 特定驗證。因此，具有CUPS 管理群組sys 和CUPS 密碼的CUPS 管理員必須在/etc/cups/passwd.md5 中輸入。要這樣做，請以root 身分輸入：

```
lppasswd -g sys -a CUPS-admin-name
```

當cupsd 以lp 身分執行時，無法產生/etc/printcap，因為lp 不被允許在/etc/ 中建立檔案。因此，cupsd 會產生/etc/cups/printcap。爲了要確保僅可從/etc/printcap 讀取佇列名稱的應用程式繼續正常運作，/etc/printcap 是指向/etc/cups/printcap 的符號連結。

當cupsd 以lp 身分執行，無法開啓連接埠631。因此，cupsd 無法以rccups reload 重新載入。請使用rccups restart。

BrowseAllow 和**BrowseDeny** 的通用功能

BrowseAllow 和BrowseDeny 設定的存取權限可套用於傳送給cupsd 的所有類型套件。/etc/cups/cupsd.conf 中的預設設定值如下：

```
BrowseAllow @LOCAL BrowseDeny All
```

以及

```
<Location />
  Order Deny,Allow
  Deny From All
  Allow From 127.0.0.1
  Allow From 127.0.0.2
  Allow From @LOCAL
</Location>
```

利用此方法，僅LOCAL 主機可存取CUPS 伺服器上的cupsd。LOCAL 主機是指屬於非PPP 介面的IP 位址之主機(是沒有設定IFF_POINTOPOINT 旗標的介面)，而其IP 位址和CUPS 伺服器屬於相同的網路。來自其他伺服器的封包會立刻被拒絕。

預設會啓用 cupsd

在標準安裝中，會自動啓用 cupsd，以便不需其他手動作業即可存取 CUPS 網路伺服器的佇列。前兩項(請參閱節 12.7.3, “cupsd 以使用者 lp 執行” 與節 12.7.3, “BrowseAllow 和 BrowseDeny 的通用功能”)是本功能的重要先決條件。如果沒有，則自動啓用 cupsd 的安全性將不足。

12.7.4 各種套件中的 PPD 檔案

僅有 PPD 檔案的印表機組態

YaST 印表機組態僅使用安裝於系統上 /usr/share/cups/model/ 中的 PPD 檔案來設定 CUPS 的佇列。爲了決定適合印表機型號的 PPD 檔案，YaST 將硬體偵測期間決定的廠商和型號與系統上 /usr/share/cups/model/ 中提供的所有 PPD 檔案內的廠商和型號相比較。基於此原因，YaST 印表機組態將從 PPD 檔案中取出的廠商和型號資訊產生資料庫。當您從廠商和型號清單中選取印表機時，會接收到相符廠商和型號的 PPD 檔案。

僅使用 PPD 檔案且不使用其他資訊來源的組態，好處在於 /usr/share/cups/model/ 中的 PPD 檔案可自由修改。YaST 印表機組態可辨識變更並重新產生廠商和型號資料庫。例如，如果您只有 PostScript 印表機，通常不需要 cups-drivers 套件中的 Foomatic PPD 檔案，或 cups-drivers-stp 套件中的 Gimp-Print PPD 檔案。您可以直接將 PostScript 印表機的 PPD 檔案複製到 /usr/share/cups/model/ (如果在 manufacturer-PPDs 套件中尚未存在)，以達到印表機的最佳組態。

cups 套件中的 CUPS PPD 檔案

cups 套件中的一般 PPD 檔案已經以 PostScript Level 1 和 Level 2 印表機適當的 Foomatic PPD 檔案補充。

- /usr/share/cups/model/Postscript-level1.ppd.gz
- /usr/share/cups/model/Postscript-level2.ppd.gz

cups-drivers 套件中的 PPD 檔案

一般情況下，Foomatic 印表機過濾器 foomatic-rip 會與非 PostScript 印表機的 Ghostscript 搭配使用。適合的 Foomatic PPD 檔案含有項目 *NickName: ... Foomatic/Ghostscript driver 與 *cupsFilter: ... foomatic-rip。這些 PPD 檔案位於 cups-drivers 套件中。

YaST 比較偏好 Foomatic PPD 檔案，如果 Foomatic PPD 檔案含有項目 *NickName: ... Foomatic ... (建議) 符合印表機型號，而 manufacturer-PPDs 套件未包含更適合的 PPD 檔案(請參閱下文)。

cups-drivers-stp 套件中Gimp-Print PPD 的檔案

除了foomatic-rip 之外，來自Gimp-Print 的CUPS 過濾器rastertoprinter 也可以用在許多非PostScript 印表機上。此過濾器和適合的Gimp-Print PPD 檔案可在cups-drivers-stp 套件中找到。Gimp-Print PPD 檔案位於/usr/share/cups/model/stp/ 中，並含有項目*NickName: ... CUPS+Gimp-Print 與*cupsFilter: ... rastertoprinter。

位於manufacturer-PPDs 套件中印表機製造廠商的PPD 檔案

manufacturer-PPDs 套件包含具有充分自由授權的印表機製造廠商所發行的PPD 檔案。PostScript 印表機應該以印表機製造廠商的適合PPD 檔案來設定，因為有此檔案才可使用PostScript 印表機的所有功能。如果符合下列條件，YaST 偏好來自manufacturer-PPDs 套件的PPD 檔案：

- 在硬體偵測期間決定的廠商和型號符合manufacturer-PPDs 套件中PPD 檔案內的廠商和型號。
- manufacturer-PPDs 套件中的PPD 檔案是唯一適合印表機型號的PPD 檔案，或是含有也符合印表機型號的*NickName: ... Foomatic/Postscript (建議) 項目的Foomatic PPD 檔案。

同時，在下列情況中，YaST 不使用任何來自manufacturer-PPDs 套件的PPD 檔案：

- 來自manufacturer-PPDs 套件的PPD 檔案不符合廠商和型號。如果manufacturer-PPDs 套件對相似型號僅包含一個PPD 檔案會發生此狀況，例如，型號系列中個別型號沒有各自的PPD 檔案，但是在PPD 檔案中以類似Funprinter 1000 series 的格式指定型號名稱。
- 不建議使用該Foomatic PostScript PPD 檔案。這可能是因為印表機型號無法在PostScript 模式下有效率的操作，例如，此模式可能不信任該印表機，因為記憶體太少或印表機處理器太弱而使得速度太慢。此外，印表機可能預設不支援PostScript，例如因為只有選用模組提供PostScript。

如果來自manufacturer-PPDs 套件的PPD 檔案不適合PostScript 印表機，但是YaST 基於上述因素無法加以設定，請在YaST 中手動選取對應印表機型號。

12.8 疑難排解

下列章節涵蓋印表機硬體和軟體最常遭遇的問題，以及解決或避免這些問題的方式。

12.8.1 沒有標準印表機語言模式支援的印表機

不支援任何常用印表機語言並僅能以特殊控制序列處理的印表機稱為GDI印表機。這些印表機僅可在製造廠商針對其開發驅動程式的作業系統版本上使用。GDI是Microsoft為繪圖設備所開發的程式設計介面。實際問題不在於程式設計介面，而是GDI印表機僅可使用對應印表機型號的專用印表機語言處理。

部分印表機可切換到GDI模式或標準印表機語言來操作。部分製造廠商提供其GDI印表機的專用驅動程式。專用印表機驅動程式的壞處在於不保證可與安裝的列印系統搭配使用，且不保證適用於各種硬體平台。相反的，支援標準印表機語言的印表機不需依賴特殊的列印系統版本或特殊硬體平台。

除了不需花費時間嘗試使專用Linux驅動程式運作，也不需花費更多成本購買支援的印表機。如此可一次解決所有驅動程式問題、減少安裝與設定特殊驅動程式軟體以及取得列印系統中新開發所需的驅動程式更新的需要。

12.8.2 PostScript 印表機沒有可用的PPD 檔案

如果manufacturer-PPDs套件不包含任何適用於PostScript印表機的PPD檔案，應該可以使用印表機製造廠商驅動程式光碟中的PPD檔案，或從印表機製造廠商的網頁下載適合的PPD檔案。

如果PPD檔案以壓縮保存檔(.zip)或自解壓縮保存檔(.exe)形式提供，請以unzip解壓縮。首先，檢閱PPD檔案的授權條款。然後使用cupstestppd公用程式檢查PPD檔案是否符合“Adobe PostScript Printer Description File Format Specification 版本4.3”。如果公用程式傳回“FAIL”，表示PPD檔案的錯誤很嚴重，可能會導致重大問題。應該要減少cupstestppd所報告的問題點。若有需要，請詢問印表機製造廠商以取得適合的PPD檔案。

12.8.3 並列埠

最安全的方法是將印表機直接連接到第一並列埠，並在BIOS中選取下列並列埠設定值：

- I/O address:378 (hexadecimal)
- Interrupt:irrelevant
- Mode:Normal, SPP, or Output Only
- DMA:disabled

如果没有這些設定值，印表機無法在並列埠上定址，請依照BIOS中的設定值，以0x378的格式在/etc/modprobe.conf中明確輸入I/O位址。如果有兩個並列埠，I/O位址分別設為378和278(十六進位)，請以0x378,0x278格式輸入。

如果没有使用中斷7，可以使用範例12.1,“/etc/modprobe.conf：第一並列埠的中斷模式”中所示的項目啓用。在啓用中斷模式之前，請檢查檔案/proc/interrupts以了解哪些中斷已經在使用中。只會顯示目前正在使用中的中斷。這可能因為作用中的硬體元件而有變化。其他任何設備都不能使用並列埠的中斷。如果您不確定，請以irq=none使用輪詢模式。

Example 12.1: /etc/modprobe.conf：第一並列埠的中斷模式

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

12.8.4 網路印表機連線方式

查明網路問題 將印表機直接連接到電腦。基於測試因素，請將印表機設為本地印表機。如果可以，問題便與網路相關。

檢查TCP/IP網路 TCP/IP網路和名稱解析必須可作用。

檢查遠端lpd 使用以下指令來測試是否可在(主機)上建立到lpd(連接埠515)的TCP連線：

```
netcat -z host 515 && echo ok || echo failed
```

如果無法建立到lpd的連線，可能是lpd不在作用中，或是有基本網路問題。

以使用者root的身分，使用以下指令來查詢(可能很長)遠端(主機)上(佇列)的狀態報告，假使對應的lpd在作用中且主機接受查詢：

```
echo -e "\004queue" \ | netcat -w 2 -p 722 host 515
```

如果lpd沒有回應，它可能不在作用中，或是有基本網路問題。如果lpd有回應，回應應該會顯示主機上的佇列為何無法列印。如果您收到像範例12.2,“來自lpd的錯誤訊息”中的回應，問題可能是因為遠端lpd所造成。

Example 12.2: 來自lpd的錯誤訊息

```
lpd: your host does not have line printer access
lpd: queue does not exist
printer: spooling disabled
printer: printing disabled
```

檢查遠端 `cupsd` 根據預設，CUPS 網路伺服器應該每隔三十秒在 UDP 連接埠 631 上廣播其佇列。同時，可使用以下指令來測試網路中是否有 CUPS 網路伺服器。

```
netcat -u -l -p 631 & PID=$!; sleep 40 ; kill $PID
```

如果廣播 CUPS 網路伺服器存在，輸出將如範例 12.3, “來自 CUPS 網路伺服器的廣播” 中所示。

Example 12.3: 來自 CUPS 網路伺服器的廣播

```
ipp://host.domain:631/printers/queue
```

可使用以下指令來測試是否可建立到〈主機〉上 `cupsd` (連接埠 631) 的 TCP 連線：

```
netcat -z host 631 && echo ok || echo failed
```

如果無法建立對 `cupsd` 的連線，則 `cupsd` 可能不是作用中或可能有基本的網路問題。假設對應的 `cupsd` 是作用中的，而且主機可以接受查詢，則 `lpstat -h host -l -t` 會傳回在〈主機〉上所有佇列的狀態報告。

此指令可用來測試〈主機〉上的〈佇列〉是否可接受由單一換行字元組成的列印工作。應該不會印出任何資料。可能會退出一張空白頁。

```
echo -en "\r" \ | lp -d queue -h host
```

網路列印或列印伺服器盒疑難排解 在列印伺服器盒中執行的暫存序列器在執行大量列印工作時，有時會造成問題。因為這是列印伺服器盒中的暫存序列器所造成，您無法解決此問題。處理方式是，透過 TCP 插槽將印表機直接連接到列印伺服器盒，以規避列印伺服器盒中的暫存序列器。請參閱節 12.5.2, “網路印表機”。

利用此方法，可減少列印伺服器盒在不同資料格式之間的轉換問題(TCP/IP 網路和本地印表機連線)。若要使用此方法，您必須知道列印伺服器盒上的 TCP 連接埠。如果印表機連接到列印伺服器盒且電源開啓，此 TCP 連接埠通常可在列印伺服器盒電源開啓一段時間之後，以 `nmap` 套件的 `nmap` 公用程式決定。例如，`nmap <IP 位址>` 會傳送列印伺服器盒的以下輸出：

Port	State	Service
23/tcp	open	telnet
80/tcp	open	http
515/tcp	open	printer
631/tcp	open	cups
9100/tcp	open	jetdirect

此輸出表示連接到列印伺服器盒的印表機可透過連接埠9100上的TCP插槽定址。根據預設，nmap僅會檢查幾個/usr/share/nmap/nmap-services中所列出一般熟知的連接埠。若要檢查所有可能的連接埠，請使用指令nmap -p <來源連接埠>-<目的連接埠> <IP位址>。這可能會花費一些時間。如需進一步資訊，請參閱nmap man頁面。

輸入以下指令

```
echo -en "\rHello\r\f" | netcat -w 1 IP-address port cat file | netcat -
```

將字元字串或檔案直接傳送到對應連接埠以測試印表機是否可在此連接埠上定址。

12.8.5 列印成品損毀而無錯誤訊息

對列印系統而言，在CUPS後端完成資料至接收者(印表機)的資料傳輸時，列印工作便完成。如果接收者的進一步處理失敗(例如，如果印表機無法列印印表機特定資料)，列印系統並不知道。如果印表機無法列印印表機特定資料，請選取更適用於印表機的不同PPD檔案。

12.8.6 停用佇列

如果到接收者的資料傳輸在數次嘗試之後完全失敗，CUPS後端(如usb或socket)會向列印系統報告錯誤(向cupsd)。後端會決定在報告資料傳輸失敗之前，是否要繼續嘗試以及要嘗試幾次是合理的。因為進一步的嘗試可能徒勞無功，cupsd會停用對應佇列的列印。除去問題的起因之後，系統管理者必須以指令/usr/bin/enable重新啟動列印。

12.8.7 CUPS 瀏覽：刪除列印工作

如果CUPS網路伺服器透過瀏覽向用戶端主機廣播它的佇列，而且在用戶端主機上有適合的本地cupsd在作用中，用戶端cupsd會從應用程式接收列印工作，並將它們轉送給伺服器上的cupsd。當cupsd接收列印工作時，會被指定新的工

作號碼。因此，用戶端主機上的工作號碼和伺服器上的工作號碼不同。因為列印工作通常會立刻轉送，所以無法以用戶端主機上的工作號碼來刪除，因為用戶端 cupsd 一旦將列印工作轉送給伺服器 cupsd，便認為列印工作已完成。

若要刪除伺服器上的列印工作，假使伺服器尚未完成列印工作(也就是，傳送給印表機)，請使用像 `lpstat -h print-server -o` 這類的指令來決定伺服器上的工作編號。使用此工作號碼，伺服器上的列印工作便可刪除：

```
cancel -h print-server queue-jobnumber
```

12.8.8 損毀的列印工作與資料傳輸錯誤

如果您在列印程序中將印表機電源關閉再打開，或是關機再重新啟動電腦，列印工作仍然在佇列中，而且會繼續列印。必須以 `cancel` 將損毀的列印工作從佇列中移除。

如果列印工作損毀，或是主機和印表機之間的通訊發生問題，印表機會印出數頁含有不明字元的紙張，因為它無法正確地處理資料。若要處理這個問題，請遵循下列步驟：

1. 若要停止列印，請從噴墨印表機取出所有紙張，或是打開雷射印表機的紙匣。高品質的印表機會有按鈕可取消目前的列印成品。
2. 列印工作可能仍在佇列中，因為只有將工作完全傳送到印表機之後，才會移除。使用 `lpstat -o` 或 `lpstat -h <列印伺服器> -o` 檢查目前佇列中的佇列。以 `cancel <佇列>-<工作編號>` (或 `cancel -h <列印伺服器> <佇列>-<工作編號>`) 刪除列印工作。
3. 即使列印工作已從佇列刪除，部份資料可能仍會傳送到印表機。請檢查對應佇列的 CUPS 後端程序是否仍在執行中，並將它終止。例如，對於連接到並列埠的印表機，可使用指令 `fuser -k /dev/lp0` 來終止仍然在存取印表機(更精確的說是並列埠)的所有程序。
4. 將印表機關閉一段時間以完全重設印表機。然後裝入紙張並開啓印表機電源。

12.8.9 除錯 CUPS 列印系統

使用以下程序以找出 CUPS 列印系統中的問題：

1. 設定 `/etc/cups/cupsd.conf` 中的 `LogLevel debug`。

2. 停止cupsd。
3. 移除/var/log/cups/error_log* 以避免必須搜尋很大的記錄檔。
4. 啓動cupsd。
5. 重覆造成問題的動作。
6. 檢查/var/log/cups/error_log* 中的訊息以辨識問題的起因。

12.8.10 若需詳細資訊

許多特定問題的解決方法在「支援資料庫」中都有說明。若您碰到印表機的問題，請參閱支援資料庫內的文章安裝印表機(*Installing a Printer*)和SUSE LINUX 9.2的印表機組態(*Printer Configuration from SUSE LINUX 9.2*)。您用關鍵字「印表機」來搜尋即可找到。

Linux 的行動計算功能

本章節提供使用Linux 行動計算功能各方面的概觀。會簡短介紹使用的各種欄位，也會說明使用硬體的基本功能。特殊需求的軟體解決方案、最大效能的選項、及使用最少電源的可能性，皆一併涵蓋在內。本章節也包含有關主題最重要資訊來源的概述。

13.1	筆記型電腦	226
13.2	行動硬體	231
13.3	行動電話和PDA	231
13.4	若需詳細資訊	232

大部份的人將行動計算的概念與筆記型電腦、PDA、和行動電話以及其彼此間資料的交換聯想在一起。本章將這個焦點延伸到行動硬體元件，如外接硬碟、隨身碟或數位相機，這些物件可以連接到筆記型電腦或桌上電腦系統。

13.1 筆記型電腦

13.1.1 筆記型電腦硬體的特殊屬性

筆記型電腦的硬體與一般桌上電腦系統不同。這是因為像可交換性、佔用的空間和省電之類標準的相關屬性。行動硬體的製造商已經發展出PCMCIA 標準(個人電腦記憶卡國際協會，*Personal Computer Memory Card International Association*)。這個標準涵蓋了記憶卡、網路介面卡、ISDN 和數據卡、以及外接硬碟。Linux 如何支援這類硬體、組態時該考慮哪些事情、哪些軟體可以用來控制PCMCIA、以及如何疑難排解可能出現的問題，都詳述於章14, PCMCIA。

13.1.2 省電

製造筆記型電腦時包含最有效使用電源系統元件，造就了它們不必靠電力系統即能使用的適用性。它們在省電的貢獻上至少與作業系統的貢獻一樣重要。SUSE LINUX 支援影響筆記型電腦電源使用的各種方法，而且對於電池電源的操作時間上產生不同的效用。下列清單以遞減排序陳述對省電的貢獻：

- 調節CPU 速度
- 暫停時關閉顯示器亮度
- 手動調整顯示器亮度
- 中斷連接已啓用但未使用的熱插件配件(USB CD-ROM、外接滑鼠、未使用的PCMCIA 卡等)
- 閒置時降低硬碟轉速

有關SUSE LINUX 中電源管理及操作YaST 電源管理模組的詳細背景資訊，請參閱章16, 電源管理。

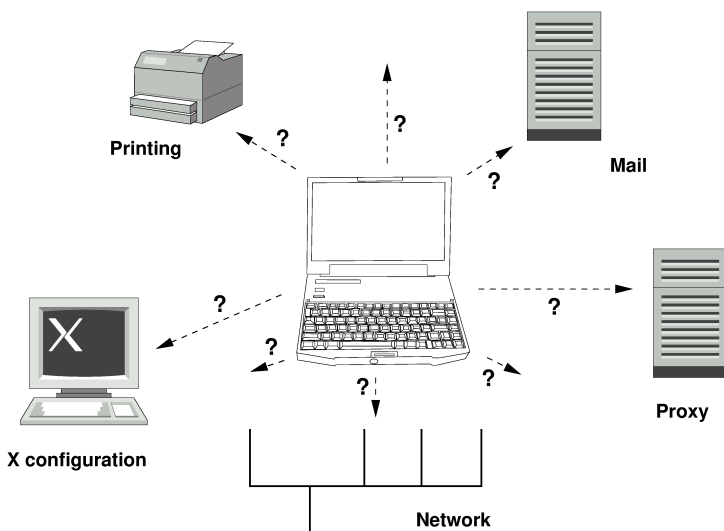


Figure 13.1: 整合網路中的筆記型電腦

13.1.3 與變動作業環境的整合

您的系統在用於行動計算時，常需與變動的作業環境搭配。必須重新設定許多依賴環境及基礎用戶端的服務，SUSE LINUX 會替您做好這項工作。

筆記型電腦在小型家用網路及公司網路之間來回交替使用的情形，影響的服務包括：

網路組態 這包括IP 位址配置、名稱解析、網際網路連線、以及與其他網路的連線。

列印 視網路而定，必須出現可用印表機及可用列印伺服器目前的資料庫。

電子郵件和Proxy 列印時，對應伺服器的清單必須是最新的。

設定X 如果您的筆記型電腦暫時連接到視訊裝置或外接螢幕，需要可用的不同顯示器組態。

SUSE LINUX 提供兩種方法，可以整合筆記型電腦到可結合的現有作業環境。

SCPM SCPM (系統組態設定檔管理, *system configuration profile management*) 允許儲存系統的任意組態狀態到一種稱為「設定檔」(*profile*)的「快照」(*snapshot*)。設定檔可以針對不同的情況而建立。當系統於變更的環境下(家用網路、公司網路)操作時,這種設定檔很有用。能夠一直在設定檔之間切換。您可以在章15,系統組態設定檔管理找到有關SCPM的資訊。KDE中的kicker applet Profile Chooser,能夠在設定檔之間切換。該應用程式切換前需要輸入root密碼。

SLP 服務位置通訊協定 (*Service Location Protocol*, SLP) 簡化了筆記型電腦到現有網路的連線。沒有SLP,筆記型電腦的管理員通常需要熟知網路中的可用服務。SLP廣播特定類型服務的可用性到本地網路中的所有用戶端。支援SLP的應用程式可以處理由SLP分派出來的資訊,而且能夠自動設定。SLP甚至能夠用來安裝系統,省略搜尋適用安裝來源的時間。您可以在章23,網路中的SLP服務找到有關SCPM的詳細資訊。

SCPM的重要性在於能啓用和維護可複製的系統情況。SLP可自動設定大部份的網路電腦組態,讓使用更為容易。

13.1.4 軟體選項

行動使用中有各種特殊工作,由專屬軟體執行:系統監控(特別是電池充電)、資料同步化、以及與周邊設備和網際網路的無線通訊。以下小節涵蓋了SUSE LINUX為每項工作提供的最重要應用程式。

系統監控

兩種KDE系統監控工具由SUSE LINUX提供。筆記型電腦可充電電池的所有狀態顯示,由kicker中的KPowerSave程式處理。複雜的系統監控由KSysguard執行。使用GNOME時,所述的功能由GNOME ACPI(作為面板程式)及系統監看提供。

KPowerSave KPowerSave是顯示控制台中可充電電池狀態的applet。圖示會調整,表示電源供應的類型。使用AC電源時,會顯示一個小型插頭圖示。使用電池時,圖示會變更為電池。要求root密碼後,會開啓電源管理的YaST模組的對應功能表。可以在其中設定不同電源供應類型下系統的行為。有關電源管理及相關YaST模組的資訊,請參閱章16,電源管理。

KSysguard KSysguard是獨立的應用程式,將系統中所有可測量的參數收集到一個監控環境中。KSysguard具有ACPI(電池狀態)、CPU負載、網路、磁碟分割及記憶體使用率的監控程式。它還能進一步監看及顯示所有系統程

序。可自訂所收集資料的顯示及篩選。透過網路可以同時監控各種資料頁的不同系統參數，或在並列埠中收集各個機器的資料。KSysguard 也能在沒有KDE 環境的機器上做為精靈來執行。有關此程式的詳細資訊，請參閱其整合的說明功能或SUSE 說明頁。

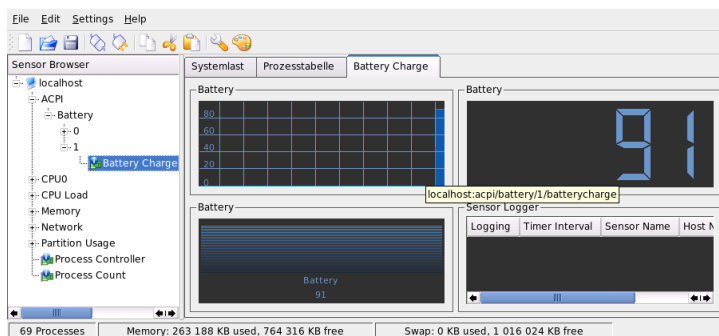


Figure 13.2: 使用KSysguard 監控電池狀態

同步化資料

在沒有連接網路的行動機器與公司中連接網路的工作站之間切換時，必須讓所有個體間已處理的資料保持同步。這些資料可能包括電子郵件資料夾、目錄以及個別檔案，在出差時需要和公司同時保持最新。這兩種情況中的解決方案如下：

電子郵件同步化 使用IMAP 帳戶在公司網路中儲存您的電子郵件。接著會從任何中斷連線的IMAP 工作站啓用電子郵件用戶端，來存取電子郵件，例如Mozilla Thunderbird Mail、Evolution 或KMail，如**用戶指南** 中所述。電子郵件用戶端需要設定，如此「傳送郵件」會永遠存取相同的資料夾。這樣可確保完成同步化程序後，所有訊息及其狀態資訊都能使用。使用郵件用戶端中執行的SMTP 服務來傳送訊息，代替全系統MTA postfix 或sendmail，來接收有關未傳送郵件的可靠回應。

同步化檔案與目錄 有數種公用程式適合用來同步化筆記型電腦與工作站之間的資料。如需詳細資訊，請參閱章31, 檔案同步化。

無線通訊

筆記型電腦不僅能夠使用電纜連接家用或公司網路，還可以透過無線方式連接其他電腦、周邊設備、行動電話或PDA。Linux 支援三種類型的無線通訊：

WLAN 涵蓋最廣泛無線技術的WLAN，是唯一適合操作大型，有時甚至是極端分散網路的模式。個別機器可以彼此連接，形成一個獨立的無線網路或存取網際網路。稱為存取點的設備，是做為啓用WLAN設備的基礎工作站，而且扮演存取網際網路的中介角色。行動使用者可以在各個存取點之間切換，端視所在位置及哪個存取點提供最佳連線而定。與行動電話的通訊方式類似，供WLAN使用者使用的大型網路，不用將它們連結到特定位置才能存取。您可以在節17.1, “無線區域網路” 找到有關WLAN的詳細資訊。

藍芽 藍芽在所有無線技術中的應用範圍最廣泛。它可以用於電腦(筆記型電腦)與PDA或行動電話之間的通訊，像IrDA的功能一樣。也可以用來連接可視範圍內的各個電腦。藍芽也應用在連接無線系統元件，像鍵盤或滑鼠。不過，此技術的範圍還不足以連接遠端系統與網路。WLAN是穿透實體障礙物(如牆)進行通訊的選用技術。如需有關藍芽、其應用程式、和組態的詳細資訊，請參閱節17.2, “藍芽”。

IrDA IrDA是最短距離的無線技術。兩邊的通訊方必須位於彼此可見的距離內。無法克服如牆之類的障礙物。IrDA的一個應用方式是從筆記型電腦傳輸檔案到行動電話。可使用IrDA來涵蓋筆記型電腦到行動電話的短距離。長距離傳送檔案到收件者是由行動網路處理。IrDA的另一種應用是在公司內以無線方式傳輸列印工作。您可以在節17.3, “紅外線資料傳輸” 找到有關IrDA的詳細資訊。

13.1.5 資料安全性

理想而言，您應使用多種方式保護筆記型電腦上的資料，不受未授權的存取。可以從下列三大面向來採取適當的安全性措施：

防止竊取 不論在什麼時候，都要避免機器遭到竊取。各種保全工具(例如鎖鏈)都可以在零售店中買到。

保全系統上的資料 重要資料在傳輸時不僅要加密，在硬碟上也要加密。這樣在竊取的情況中可確保其安全。使用SUSE LINUX建立加密磁碟分割的內容，請參閱節34.3, “加密分割區和檔案”的說明。

網路安全性 應該保護任何資料的傳輸，無論其方式為何。有關Linux及網路的一般安全性議題，請參閱節34.4, “安全性與機密性”。與無線網路相關的安全性措施，請參閱章17, 無線通訊。

13.2 行動硬體

SUSE LINUX 能夠自動偵測 Firewire (IEEE 1394) 或 USB 上的行動儲存設備。行動儲存設備這個詞，包含了任何種類的 Firewire 或 USB 硬碟、USB 隨身碟或數位相機。只要這些設備一透過對應介面與系統連接，即可經由熱插自動偵測與設定。subfs 和 submount 確保設備裝載到檔案系統中的對應位置。使用者完全不用手動裝載和卸載舊版 SUSE LINUX 找到的設備。只要程式不需存取設備，即可直接拔除。

外接磁碟(USB 和 Firewire) 只要系統正確辨認到外接磁碟，外接磁碟的圖示就會出現在‘我的電腦’(KDE) 或‘電腦’(GNOME) 中已裝載設備的清單內。在圖示上按一下滑鼠左鍵，會顯示設備的內容。在此可以建立、編輯或刪除資料夾及檔案。若要重新命名磁碟的系統指定名稱，請在圖示上按一下滑鼠右鍵所開啓的功能表中，選取對應的功能表項目。此名稱的變更，僅限顯示於檔案管理員中。因此，裝載於 /media/usb-xxx 或 /media/ieee1394-xxx 中設備的描述符號不會受影響。

USB 隨身碟 這些設備由系統處理，如外接硬碟一樣。同樣可以在檔案管理員中重新命名項目。

數位相機(USB 和 Firewire) 由系統辨識的數位相機，在檔案管理員綜覽中也是顯示為外接磁碟機。KDE 允許讀取及存取 URL camera:/ 上的圖片。而且影像可以使用 digikam 或 The GIMP 進行處理。使用 GNOME 時，Nautilus 會將圖片顯示在自己的資料夾中。簡單的影像處理及管理公用程式是 GThumb。進階的相片處理則是使用 The GIMP。除了 GThumb 之外，*用戶指南* 中均有這些程式的說明。另外也有關於數位相機的章節。

行動資料磁碟機的安全

行動硬碟或隨身碟像筆記型電腦一樣容易竊取。建議如節 34.3, “加密分割區和檔案” 中所述建立加密磁碟分割，防止其他人濫用。

13.3 行動電話和 PDA

桌上電腦或筆記型電腦可以透過藍芽或 IrDA 進行通訊。有些型號支援兩種通訊協定，而有些則僅支援其中一種。兩種通訊協定的使用範圍及相關的延伸說明文

件已經在節13.1.4, “無線通訊”中提及。行動電話通訊協定的組態, 於手冊中均有說明。有關Linux 端的組態, 於節17.2, “藍芽”和節17.3, “紅外線資料傳輸”中有詳細的說明。

Evolution 與Kontact 內建的同步化已經支援Palm, Inc. 生產的掌上型設備。開始與設備連線後, 不論使用哪種方式, 都可輕鬆透過精靈的協助加以使用。設定Palm Pilot 的支援後, 必須判斷應該同步化哪些類型的資料(通訊錄、約會等)。這兩種群組軟體應用程式, 詳述於*用戶指南*中。

整合於Kontact 程式的KPilot, 也可當作獨立公用程式使用。此內容詳述於*用戶指南*。KitchenSync 程式也可以用來同步化通訊錄資料。

您可以在*用戶指南*中找到有關Evolution 和Kontact 的詳細資訊。

13.4 若需詳細資訊

有關行動設備與Linux 所有問題的參考重點, 請參閱<http://tuxmobil.org/>。該網站的各個小節, 能夠處理筆記型電腦、PDA、行動電話及其他行動硬體的硬體及軟體方面。

<http://tuxmobil.org/> 的類似方法由<http://www.linux-on-laptops.com/> 提供。在此可以找到有關筆記型電腦和掌上型設備的資訊。

關於筆記型電腦相關議題, SUSE 提供了專屬的郵件清單(使用德文)。請參閱<http://lists.suse.com/archive/suse-laptop/>。在此清單上, 使用者與開發人員討論了SUSE LINUX 行動計算的所有觀念。以英文張貼的文章會收到回應, 但是大部分的歸檔資訊僅有德文版本。

在筆記型電腦上使用SUSE LINUX 管理電源的問題, 建議參閱/[usr/share/doc/packages/powersave](#) 中的檔案。這個目錄中通常包含測試者及開發人員的最新回應, 因此對解決問題提供了寶貴的提示。

PCMCIA

本節包含一些膝上型電腦中與PCMCIA 硬體和軟體特別相關的觀念。PCMCIA 是國際個人電腦記憶卡協會(*Personal Computer Memory Card International Association*) 的英文縮稱，為相關軟硬體的統稱。

14.1	硬體	234
14.2	軟體	234
14.3	組態	235
14.4	公用程式	237
14.5	疑難排解	237
14.6	若需詳細資訊	239

14.1 硬體

PCMCIA 卡是首要元件，有下列兩種類型：

PC 卡 是PCMCIA 最初所使用的卡片。它們使用16 位元匯流排來傳送資料，價格非常低廉。目前有些PCMCIA 橋接器不易偵測到這類的卡片。不過，一旦偵測得到，通常就會運作順暢，不會發生任何問題。

CardBus 卡 這是較新的標準，使用了32 位元匯流排，因而速度較快，但也較為昂貴。它們整合在像是PCI 卡的系統中，運作十分順暢。

如果啓動了PCMCIA 服務，您可以使用`cardctl ident` 指令來顯示插入的卡片類型。在`/usr/share/doc/packages/pcmcia` 目錄的`SUPPORTED.CARDS` 檔案中有支援卡片的清單。最新版本的PCMCIA HOWTO 也在此目錄中。

第二重要的元件是PCMCIA 控制器、PC 卡或是CardBus 橋接器，這些元件可以建立卡片與PCI 匯流排的連線。所有常見型號皆能支援。可以使用`pcic-probe` 指令來判斷控制器的類型。如果是PCI 設備，則使用`lspci-vt` 指令可以得到更進一步的資訊。

14.2 軟體

14.2.1 基本模組

必要的核心模組位於核心套件中。此外，還需要`pcmcia` 套件和`Hotplug` 套件。啓動PCMCIA 時，分別會載入`pcmcia_core`、`yenta_socket` 以及`ds` 模組。在少數的情況下，需要的是`tcic` 模組而非`yenta_socket` 模組。這些模組能啓始現有的PCMCIA 控制器，並提供基本功能。

14.2.2 卡片管理員

因爲可在系統執行時變更PCMCIA 卡，所以必須監控插槽的任何變動。這項工作是由基本模組中的卡片服務來處理。插入卡片的啓始化是由卡片管理員(如果是PC 卡)來處理，或由核心的`Hotplug` 系統(如果是CardBus 卡)來處理。在載入基本模組後，PCMCIA 啓動程序檔將啓動卡片管理員。`Hotplug` 則會自動啓動。

插入卡片之後，卡片管理員或`Hotplug` 會判斷其類型與功能並載入相關模組。在成功載入模組之後，卡片管理員或`Hotplug` 會視卡片的功用來啓動特定的啓始化

程序檔。程序檔可建立網路連線、裝載外接式SCSI硬碟的分割區，或執行其他特定硬體動作。卡片管理員的這類程序檔位於`/etc/pcmcia`目錄中。Hotplug程序檔則位於`/etc/hotplug`中。移除卡片之後，卡片管理員或Hotplug會使用相同的程序檔來終止所有卡片的活動。接著將卸載不再使用的模組。

這些動作都算是Hotplug事件。新增硬碟或分割區時(區塊事件)，Hotplug程序檔會使用`subfs`來讓新媒體可立即在`/media`中使用。如果要利用舊的PCMCIA程序檔來裝載媒體，應停用Hotplug中的`subfs`。

PCMCIA的啟動與卡片事件都會記錄在系統的記錄檔案(`/var/log/messages`)中。曾經載入的模組與執行過的程序檔都會記錄在記錄檔案中。

理論上，移除PCMCIA卡不需任何其他動作。在沒有其他作用中的網路連線時，網路卡、數據卡以及ISDN卡皆可順利移除。但這不適用在外接式硬碟或NFS目錄的裝載分割區。這些單位應以適當的程序來同步處理和解除裝載。一旦卡片除，當然無法執行適當程序。如果不能確定，請使用`cardctl eject`指令，關閉所有仍插入於筆記型電腦的卡片。如果只要關閉其中某一張卡片，請指定插槽號碼，例如，`cardctl eject 0`。

14.3 組態

您可以使用YaST Runlevel編輯器，決定是否要在系統開機時啟動PCMCIA。請使用‘系統’→‘Runlevel編輯器’來開啓此模組。

下列三個變數定義於`/etc/sysconfig/pcmcia`檔案中：

PCMCIA_PCIC 包含了控制PCMCIA控制器的模組的名稱。啟動程序檔通常應能自動判定模組。如果失敗，請在此輸入模組。否則，此變數會留下空白。

PCMCIA_CORE_OPTS 此變數是為`pcmcia_core`模組的參數所設計。不過，不常用到這些變數。在`pcmcia_core(4)`手冊頁中有對此選項做說明。因為此手冊頁會提到David Hinds的`pcmcia-cs`套件中的同名模組，其中列出的參數比核心實際支援的模組還多，核心支援模組是指那些以`cb_`和`pc_debug`為名稱開頭的參數。

PCMCIA_BEEP 啓用和關閉卡片服務員的音效訊號。

`/etc/pcmcia/config`檔案與`/etc/pcmcia/*.conf`檔案皆包含PC卡的驅動程式的設定內容。首先，會讀取`config`，接著按照字母順序來讀取`/*.conf`。最後會使用所找到的最後一個卡片項目。有關這些檔案的語法詳細資訊，可以在`pcmcia(5)`手冊頁中找到。

`/etc/sysconfig/hardware/hwcfg-<>` 指定的檔案含有CardBus卡的驅動程式的設定內容。設定卡片時，YaST會建立這些檔案。更多關於組態名稱的資訊，在`/usr/share/doc/packages/sysconfig/README`中以及`getcfg(8)`手冊頁中都可以找到。

14.3.1 網路卡

如同一般的網路卡，乙太網路、無線區域網路以及記號環網路卡都可以使用YaST來設定。如果偵測不到卡片，請在硬體設定中選取PCMCIA卡片類型。其他有關網路組態的詳細資訊，在節22.4, “使用YaST手動設定網路連線”中可以找到。

14.3.2 ISDN

如同其他ISDN卡，ISDN PC卡可利用YaST來設定至較大的傳輸量。選取清單中的哪一片PCMCIA ISDN卡並不要緊，只要是PCMCIA卡即可。在設定硬體和選取提供者時，作業模式必須是Hotplug，而不是開機時啟動。ISDN數據機也有PCMCIA卡的形式。這些是數據卡或附帶有ISDN連線套件的多功能卡。它們都被當成和數據機一樣。

14.3.3 數據機

通常數據PC卡不會有PCMCIA專用的設定。數據機一旦插入，即可在`/dev/modem`找到它。有些PCMCIA數據卡為軟體數據機，Linux並不支援。如果能夠取得這些卡的驅動程式，應將它們安裝在系統中。

14.3.4 SCSI 與IDE

您可以透過卡片管理員或Hotplug來載入適合的驅動程式模組。SCSI卡或IDE卡一旦插入，即可使用連接的設備。此設備的名稱是動態決定的。關於可用的SCSI設備與IDE設備的資訊，位於`/proc/scsi`以及`/proc/ide`之下。

外接式硬碟、CD-ROM光碟機以及類似的設備，在將PCMCIA卡插入插槽前，都必須先行開啓電源。SCSI設備請使用主動式終端模式。

┌

移除SCSI 或IDE 卡

在移除SCSI 或IDE 卡前，必須以umount 指令解除裝載所有連接設備的分割區。如果您忘記這麼做，只有再重新啓動系統後，才能存取這些設備。

└

14.4 公用程式

上述的cardctl 公用程式，是取得PCMCIA 資訊或執行一些動作的主要工具。若需詳細資訊，請參閱cardctl(8) 的手冊頁。請輸入cardctl 以取得有效選項清單。圖形化的前端cardinfo 可用來控制cardctl 的主要功能。如果要使用圖形化的前端，請安裝pcmcia-cardinfo 套件。

ifport、ifuser、probe 以及rcpcmcia 是一些在pcmcia 套件中的其他公用程式。不過，不常用到它們。如果要找出pcmcia 套件包含哪些檔案，請輸入rpm-ql pcmcia 指令

14.5 疑難排解

如果在筆記型電腦或一些卡片中，發生PCMCIA 方面的問題，多半都能按步就班地找出問題並解決問題。首先，找出是因為卡片而造成問題，或是因為PCMCIA 基本系統而造成問題。如果要這麼做，請在不插入任何卡片的情況下啓動電腦。如果基本系統運行順暢，則插入卡片。所有訊息都會記錄在/var/log/messages 中。在搜尋問題原因時，請使用tail-f /var/log/messages 監控此檔案。使用此方法，可將問題縮小至下列兩種情況之一。

14.5.1 PCMCIA 基本系統無法運作

如果系統中止於此訊息PCMCIA：啓動服務或在啓動系統時發生其他問題，請重新啓動系統並在開機提示中輸入NOPCMCIA=yes 來關閉PCMCIA。要找出錯誤，請以手動方式將PCMCIA 系統的三個基本模組一一載入。

如果要手動載入PCMCIA 模組，請以root 使用者的身份，執行modprobe pcmcia_core、modprobe yenta_socket 以及modprobe ds 等指令。在少數

的情況下，必須使用`tcic`、`i82365` 或 `i82092`，而非 `yenta_socket`。先載入的兩個模組就是關鍵模組。

如果是在載入 `pcmcia_core` 時發生錯誤，請參閱 `pcmcia_core(4)` 手冊頁。您可以先使用 `modprobe` 指令來測試此手冊頁中所提的選項。例如，可以考慮對可用的 I/O 區域做測試。有時這種測試會干擾其他硬體元件，並因此造成錯誤。可以使用 `probe_io=0` 選項來避免這種錯誤：

```
modprobe pcmcia_core probe_io=0
```

如果選取的選項成功，請在 `/etc/sysconfig/pcmcia` 檔案中設定 `PCMCIA_CORE_OPTS` 變數，將值設為 `probe_io=0`。使用空格來分隔多個選項：

```
PCMCIA_CORE_OPTS="probe_io=0 setup_delay=10"
```

如果是在載入 `yenta_socket` 模組時發生問題，則表示是在更基本的地方出現問題，例如 `ACPI` 的資源配置問題。

`/etc/pcmcia/config` 檔案與 `/etc/pcmcia/config.opts` 檔案是由卡片管理員來做解譯。其中的設定，一部份是用來啟動 `cardmgr`，一部份用來載入 PC 卡的驅動程式模組。`IRQ`、`I/O` 連接埠以及記憶位址範圍可包含在 `/etc/pcmcia/config.opts` 檔案內或排除在外。在少數的情況下，存取錯誤的 I/O 區域會導致系統損毀。如果出現這種情況，請試著限制這些區域。

14.5.2 PCMCIA 卡無法正常運作

基本上，會有下列三種類型的錯誤：偵測不到卡片、無法載入驅動程式或驅動程式所提供的介面設定錯誤。您必須知道此卡片是否由卡片管理員控制或是由 `Hotplug` 控制。卡片管理員控制 PC 卡，`Hotplug` 控制 `CardBus` 卡。

插入卡片時沒有反應。如果系統在插入卡片時沒有反應，且手動執行 `cardctl insert` 指令也同樣沒有幫助，則 `PCI` 設備的中斷配置可能有誤。如果出現這種情況，其他如網路卡等 `PCI` 設備也有可能出錯。遇到這種情形，可以使用 `pci=noacpi` 開機參數或其他的 `PCI` 或 `ACPI` 參數。

偵測不到卡片 如果偵測不到卡片，`/var/log/messages` 中會出現不支援 `x` 插槽中的卡片的訊息。此訊息僅表示卡片管理員無法指定驅動程式給該卡片。需使用 `/etc/pcmcia/config` 檔案或 `/etc/pcmcia/*.conf` 檔案，以供指定。透過將現有的選項作為樣板，可輕易擴充此驅動程式資料庫。您可以輸入 `cardctl ident` 指令來找出有關卡片的詳細資訊。更多與此相關資訊可在 `PCMCIA HOWTO` (第 6 小節) 與 `pcmcia(5)` 手冊頁中找到。在編輯 `/etc/pcmcia/config` 或 `/etc/pcmcia/*.conf` 後，請使用 `rcpcmcia reload` 指令來重新載入驅動程式。

未載入驅動程式 可能原因之一是驅動程式資料庫內含錯誤的指定。例如，可能是因為製造商在相同的卡片型號中使用不同的晶片。有些型號僅適用或較適用於部分驅動程式，而不適合使用預先選取的驅動程式。遇到這種情形，則需取得該卡片的詳細資訊。必要時，請將問題張貼在郵件清單中，或是詢問「進階支援」部門

就CardBus卡而言，則應將HOTPLUG_DEBUG=yes項目插入於/etc/sysconfig/hotplug檔案中。接下來檢查系統記錄的訊息，找出是否已成功載入驅動程式。

另一個可能原因是資源衝突。就大多數的PCMCIA卡而言，使用了哪個IRQ、I/O連接埠或記憶體區域並不要緊，但仍會有例外。如果出現這種情況，則一張張地測試每張卡片，並暫時關閉其他系統元件，例如音效卡、IrDA、數據機以及印表機等元件。以root使用者的身份使用lsdev指令來檢視資源配置。如果有多个PCI設備使用同一個IRQ時，此方法十分有效。

有個可能的解決方案是找出一個適用於卡片驅動程式模組的選項。輸入modinfo (driver)以列出選項。多數模組會提供手冊頁。rpm -ql pcmcia | grep man則會列出pcmcia套件所包含的所有手冊頁。如果要測試這些選項，您可以手動卸載卡片驅動程式。

一旦您已找到解決方案，一旦找出一個解決方案，通常您可以在/etc/pcmcia/config.opts中允許或禁止某些資源的使用。您也可以在此檔案中輸入卡片驅動程式的選項。例如，若要單使pcnet_cs模組獨利用IRQ 5，可加入下列設定：

```
module pcnet_cs opts irq_list=5
```

介面設定錯誤 遇到這種情形，請使用getcfg謹慎地檢查介面的設定以及設定的名稱以排除設定錯誤。/etc/sysconfig/network/config檔案中的DEBUG變數以及/etc/sysconfig/hotplug檔案中的HOTPLUG_DEBUG變數都應設為yes。如果這方法無效或是要針對其他卡片，您可以在卡片管理員或Hotplug所執行的程序檔中加入set-vx(請見/var/log/messages)。在此方法下，程序檔中的每一個指令都會記錄在系統記錄中。如果您在程序檔中找到關鍵區段，請在終端機中輸入和測試個別的指令。

14.6 若需詳細資訊

若需有關特定筆記型電腦模型的實用資訊，請參閱Linux筆記型電腦首頁<http://linux-laptop.net>。位於<http://tuxmobil.org/>的TuxMobil

首頁，亦提供有用的資訊。這些頁面提供筆記型電腦HOWTO、IrDA HOWTO 以及其他有用的資訊。此外，位於<http://portal.suse.com> 的支援資料庫則擁有多篇文章，告訴您在行動設備中使用Linux 的方法。若需這些文章，請在搜尋對話方塊中輸入*notebook* 關鍵字或*laptop* 關鍵字。

系統組態設定檔管理

本章將介紹SCPM (系統組態設定檔管理)。透過SCPM 的協助，可將您電腦的組態與不同作業環境或硬體組態搭配。SCPM 管理不同狀況的一組系統設定檔。SCPM 讓您可以輕易在兩個系統設定檔之間切換，減少手動重新設定系統的需要。

15.1	詞彙	242
15.2	使用指令行設定SCPM	242
15.3	YaST 設定檔管理員	245
15.4	疑難排解	248
15.5	在系統開機時選取設定檔	249
15.6	若需詳細資訊	249

有些情況要求修改的系統組態。大部份狀況是在不同位置操作的行動電腦。如果桌面系統要暫時使用與平常不同的硬體元件操作，SCPM 就非常好用。還原原始系統組態會變得很容易，而且可以重新產生系統組態的修改。使用SCPM，系統組態的任何部份均可保存在自訂的設定檔中。

SCPM 應用程式的主要欄位是膝上型電腦的網路組態。不同的網路組態通常要求其他服務的不同設定，例如電子郵件和Proxy。然後接著是其他元件，如不同的家用和公司用印表機、會議中的多媒體投影機之自訂X 伺服器組態、路途上的特殊省電設定，或海外分公司的不同時區。

15.1 詞彙

下列為SCPM 文件和YaST 模組中常用的詞彙。

- 詞彙系統組態代表電腦的完整組態。它涵蓋所有基礎設定，如硬碟分割區的使用、網路設定、時區的選擇和鍵盤對應。
- 設定檔也稱為組態設定檔，是將狀態保留起來，可以隨時還原。
- 作用中的設定檔代表最後選取的設定檔。這並不表示目前的系統組態精確地對應到此設定檔，因為可隨時修改組態。
- 在SCPM 內容中，資源是提供系統組態的元件。可以是包括中繼資料(如使用者、權限或存取時間)的檔案或軟式連結(softlink)。也可以是在此設定檔中執行，但是在另一個設定檔中停用的系統服務。
- 每一個資源皆屬於某個特定的資源群組。這些群組包含邏輯上屬於一起的資源，而大部份群組同時包含服務和其組態檔。組合SCPM 所管理的資源是很容易的，因為不需要想要服務之組態檔案的任何知識。SCPM 和預先設定的資源群組選項一起提供，對於大部份狀況應該已經夠用。

15.2 使用指令行設定SCPM

本節僅介紹SCPM 的指令行組態。瞭解如何啟動它、設定它和使用設定檔。

15.2.1 啓動SCPM 與定義資源群組

SCPM 必須在使用前啓用。以 `scpmenable` 啓用 SCPM。第一次執行時，SCPM 會花費數秒鐘起始化。以 `scpm disable` 隨時停用 SCPM 以避免意外切換設定檔。後續的重新啓動只會繼續起始化。

根據預設，SCPM 會處理網路和印表機設定以及 X.Org 組態。若要管理特殊服務或組態檔，請啓用對應的資源群組。若要列出預先定義的資源群組，請使用 `scpmlist_groups`。若僅要查看已經啓用的群組，請使用 `scpmlist_groups -a`。請在指令行以 `root` 的身份執行這些指令。

```
scpm list_groups -a

nis                Network Information Service client
mail               Mail subsystem
ntpd               Network Time Protocol daemon
xf86               X Server settings
autofs             Automounter service
network            Basic network settings
printer            Printer settings
```

以 `scpmactivate_group NAME` 或 `scpmdeactivate_group NAME` 啓用或停用群組。以相關群組名稱取代 `NAME`。

15.2.2 建立與管理設定檔

在啓用 SCPM 之後，名為 `default` 的設定檔就已經存在。以 `scpm list` 取得所有可用設定檔的清單。此一現有的設定檔也是作用中的設定檔，可使用 `scpm active` 來確認。設定檔 `default` 是基本組態，其他設定檔均從此衍生。因此，應該先進行在所有設定檔中均相同的設定。然後將這些修改以 `scpm reload` 儲存在作用中的設定檔中。`default` 設定檔可複製並重新命名作為新設定檔的基礎。

有兩種方法可新增設定檔。如果新設定檔(在此處名為 `work`)要以設定檔 `default` 為基礎，請以 `scpmcopy default work` 建立它。指令 `scpmswitch work` 可變為新設定檔，然後便可修改。您可能基於某些特殊原因想要修改系統組態，以及將變更儲存到新設定檔。指令 `scpmadd work` 可藉由將目前系統組態儲存在設定檔 `work` 中，並將它標示為作用中來建立新設定檔。執行 `scpm reload`，然後將變更儲存到 `work` 設定檔。

可使用指令 `scpmrename x y` 和 `scpmdelete z` 來重新命名或刪除設定檔。例如，若要將 `work` 重新命名為 `project`，請輸入 `scpmrename work project`。若要刪除 `project`，請輸入 `scpm delete project`。作用中的設定檔無法刪除。

15.2.3 切換組態設定檔

`scpm switch work` 指令可切換到另一個設定檔(在此例中為`work` 設定檔)。切換到作用中的設定檔可將系統組態修改的設定包括在設定檔中。這等同於指令 `scpm reload`。

在切換設定檔時，SCPM 首先檢查作用中設定檔的哪些資源被修改。然後查詢每一個資源的修改是否應新增到作用中的設定檔或捨棄。如果您偏好分別列出資源(如先前版本的SCPM)，請以 `-r` 參數使用切換指令：`scpm switch -r work`。

```
scpm switch -r work
```

```
Checking for modified resources
Checking for Resources to be started/shut down
Checking for dependencies
Restoring profile default
```

然後SCPM 會將目前系統組態與要切換的設定檔相比較。在此階段中，SCPM 會評估因為相互依存性或反映組態中變更而要停止或重新啓動的系統服務。這就像僅涉及小部份系統的部份系統重新開機，而沒有變更的其他部份仍可繼續運作。只有在此時系統服務會停止、所有修改的資源(如組態檔案)會寫入，而系統服務會重新啓動。

15.2.4 進階設定檔設定

您可以對每一個以 `scpmlist` 顯示的設定檔輸入描述。對於作用中的設定檔，以 `scpm set description "文字"` 設定。例如，以 `scpm set description "text" work` 提供非作用中設定檔的名稱。有時您可能想在切換設定檔時執行SCPM 未提供的其他動作。最多可對每一個設定檔附加四個執行檔。可在切換過程的不同階段呼叫。這些階段有：

prestop 當離開設定檔時，在停止服務之前

poststop 當離開設定檔時，在停止服務之後

prestart 當啓用設定檔時，在啓動服務之前

poststart 當啓用設定檔時，在啓動服務之後

輸入 `scpm set prestop filename \scpm set poststop filename \scpm set prestart filename` 或 `scpm set poststart filename` 以 `set` 指令插入這些動作。程序檔必須為執行檔，並參考正確的解譯器。

┌

整合自訂程序檔

其他可由SCPM執行的程序檔必須為超級使用者(root)可讀取且可執行。必須對其他所有使用者封鎖這些檔案的存取權。輸入`chmod 700 filename`與`chown root:root filename`以給予root對檔案的專用許可權。

└

以`set`與`get`查詢所輸入的其他所有設定。例如，`scpm get poststart`指令可傳回`poststart`呼叫的名稱，而若沒有附加則不會傳回任何內容。以`" "`覆寫來重設此設定。`scpm set prestop " "`指令會移除附加的`prestop`程式。

所有`set`和`get`指令均可使用新增註解的方式套用於任意設定檔。例如，`scpm get prestop filename work`或`scpm get prestop work`。

15.3 YaST 設定檔管理員

透過YaST控制中心(系統 → 設定檔管理員)啟動YaST設定檔管理員。在一開始啟動時，選取在‘SCPM選項’對話方塊中的‘啓用’，如圖形15.1, “YaST SCPM選項”中所示。在‘設定’中，決定進度快顯視窗是否應該自動關閉，以及是否顯示SPCM組態進度的詳細資訊。‘切換模式’可以在切換設定檔時，決定是否應該儲存或放棄作用中設定檔的已修改資源。如果‘切換模式’設為‘一般’，在切換時會儲存作用中設定檔中的所有變更。若要在開機時定義SCPM的行為，請將‘開機模式’設為‘儲存變更’(預設值)或是‘放棄變更’。

15.3.1 設定資源群組

若要對目前的資源組態進行變更，請選擇在‘SCPM選項’對話方塊中的‘設定資源’。下一個對方塊‘資源群組的組態’(如圖形15.2, “資源群組組態”中所示)，會列出系統上所有可用的資源。若要新增或編輯資源群組，請指定或修改‘資源群組’與‘描述’。以LDAP服務為例，請輸入`ldap`做為‘資源群組’與LDAP用戶端服務以做為‘描述’。然後輸入適當的資源(服務、組態檔或兩者)或修改現有的資源。刪除不使用的資源。若要重設選取資源的狀態(放棄對它們所做的任何變更，並還原成起始的組態值)，請選擇‘重設群組’。您的變更會儲存至作用中的設定檔。

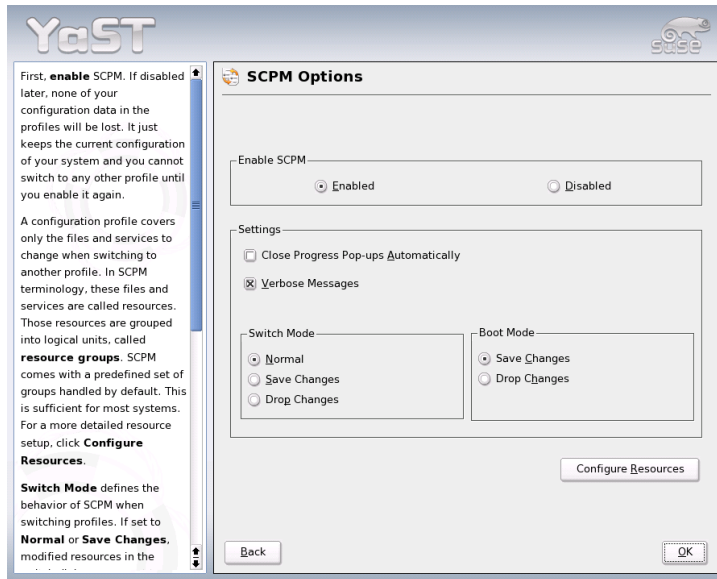


Figure 15.1: YaST SCPM 選項

15.3.2 建立新設定檔

若要建立新的設定檔，請按一下啓動對話方塊中的‘新增’（系統組態設定檔管理）。在開啓的視窗中，選取新設定檔是否應該根據目前的系統組態(SCPM 會自動擷取目前的組態並將它寫入設定檔)或根據現有的設定檔。如果您使用目前的系統組態做爲新設定檔的基礎，您可以將新設定檔標示爲新作用中的設定檔。這並不會變更舊設定檔，而且也不會啓動或停止任何服務。

在下列對話方塊中提供新設定檔的名稱與簡短描述。若是SCPM 要在切換設定檔時執行特殊的程序檔，請輸入每個執行檔的路徑(請參閱圖形 15.3, “設定檔特殊設定”)。請參閱節 15.2.4, “進階設定檔設定”，以取得更多的資訊。SCPM 將針對新設定檔的資源進行檢查。在成功地完成此測試後，就可以開始使用新的設定檔。

15.3.3 修改現有設定檔

若要修改現有的設定檔，請選擇啓動對話方塊中的‘編輯’（系統組態設定檔管理）。然後依據需求修改名稱、描述、程序檔以及資源。

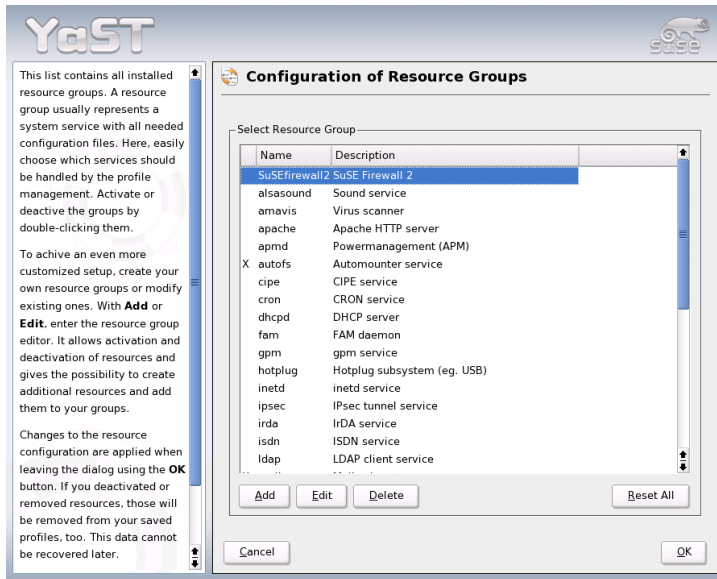


Figure 15.2: 資源群組組態

15.3.4 切換設定檔

若要切換設定檔，請開啓設定檔管理員。作用中的設定檔是以箭頭標示。選取要切換的設定檔，然後按一下‘切換到’。若要需要的話，SCPM 會檢查新的或修改過的資源，然後新增它們。

如果已修改資源，YaST 會開啓‘確認切換’對話方塊。‘作用中設定檔的已修改資源群組’會列出已修改的作用中設定檔的所有資源群組，但不會儲存到作用中設定檔。目前選取的資源群組的‘儲存或忽略’，可以決定是否應該將此資源群組的變更儲存到作用中設定檔或者放棄變更。也可以選取每個資源，然後按一下‘詳細資訊’以詳細分析變更。這將顯示出屬於此已修改資源群組的所有組態檔或執行檔的清單。若要取得舊版與新版的逐行比較，請按一下‘顯示變更’。在分析變更後，請在‘動作’中決定要如何處理它們。

儲存資源 將此資源儲存到作用中的設定檔，但是讓所有其他的設定檔維持不變。

忽略資源 使作用中資源維持不變。此變更已放棄。

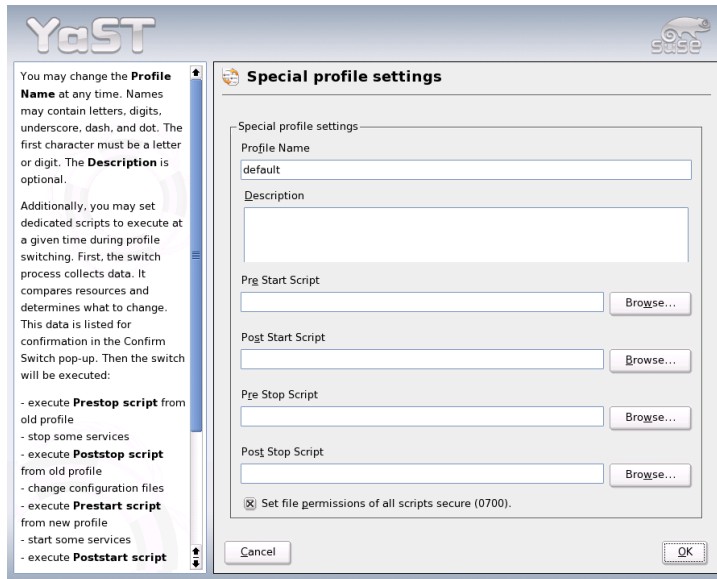


Figure 15.3: 設定檔特殊設定

儲存到所有設定檔 複製此資源的整個組態到所有其他的設定檔。

修補所有設定檔 僅套用最近的變更至所有的設定檔。

‘儲存或忽略全部’ 僅儲存或放棄此對話方塊中顯示的所有資源變更。

在確認作用中設定檔的變更後，請按一下‘確定’以結束‘確認切換’對話方塊。SCPM 接著會切換至新的設定檔。在切換時，它會執行舊設定檔的預先停止程序檔以及稍後停止程序檔，以及新設定檔的預先啟動程序檔以及稍後啟動程序檔。

15.4 疑難排解

本節涵蓋SCPM 常遭遇的問題。請瞭解它們是如何產生的以及如何才能解決這些問題。

15.4.1 在切換過程中終止

有時SCPM 在切換程序期間會停止工作。這可能是某些外部效果所造成，例如：使用者中止、電源中斷，或甚至SCPM 本身的錯誤。如果發生，在下次啓動SCPM 時會顯示錯誤訊息表示SCPM 已鎖定。這是基於系統安全，因為儲存在其資料庫中的資料可能和系統狀態不同。若要解決此問題，請執行 `scpm recover`。SCPM 會執行前一次執行的所有遺失作業。您也可以執行 `scpm recover -b`，它嘗試恢復前一次執行的所有已執行作業。如果您正在使用YaST 設定檔管理員，將會在啓動時復原對話方塊，以執行上方所述的指令。

15.4.2 變更資源群組組態

若要在SCPM 已經起始化時修改資源群組的組態，請在新增或移除群組之後輸入 `scpmrebuild`。利用此方法，新資源可新增到所有設定檔，而移除的資源會永久刪除。如果刪除的資源在不同設定檔中設定不同，此組態資料會遺失(除了系統中目前的版本之外，因為SCPM 不會接觸到它)。如果您以YaST 修改組態，不需要輸入重建指令，因為YaST 會處理。

15.5 在系統開機時選取設定檔

若要在系統開機時選取設定檔，請在開機畫面時按下 **(F4)** 以存取可用設定檔的清單。使用方向鍵選取設定檔並以 **(Enter)** 確認選擇。然後選取的設定檔便可作為開機選項使用。

15.6 若需詳細資訊

最新的文件可在SCPM 資訊頁面中找到，您可以使用Konqueror 或Emacs (`konqueror info:scpm`) 等工具來檢視。在主控台中，輸入 `info` 或 `pinfo`。給開發人員的資訊可在 `/usr/share/doc/packages/scpm` 中找到。

電 源 管 理

本章會針對Linux 的各種電源管理技術提供簡介，並詳細說明所有APM (進階電源管理) 的組態、ACPI (進階組態與電源介面) 的組態以及CPU 頻率比例的設定。

16.1	省電功能	252
16.2	APM	253
16.3	ACPI	254
16.4	硬碟的休眠	259
16.5	powersave 套件	260
16.6	YaST電源管理模組	267

APM 之前只用於筆記型電腦的電源管理上，而硬體資訊與組態工具ACPI 則可用於所有目前的電腦中(筆記型電腦、桌上型電腦及伺服器)。在許多目前硬體的類型中，CPU 頻率可視狀況調整，這有助於節省電池使用時間，尤其是在行動設備上(CPU 頻率比例)。

電源管理技術都需要合適的硬體與BIOS 常式。多數的筆記型電腦與目前許多桌上型電腦及伺服器都符合這項需求。過去許多電腦都是使用APM。由於APM 大部份是由實作在BIOS 中的一組功能所組成，使得APM 支援的等級會因硬體而異。ACPI 更加有此特性，其組成也更加複雜。因此，實際上是無法說明哪種技術比較好。只有對硬體進行各種測試，然後再選出支援性最好的技術。

AMD64 處理器的電源管理技術

AMD64 處理器具有64 位元的核心，僅支援ACPI。

16.1 省電功能

省電功能不僅對於筆記型電腦的行動用途很重要，對於桌上型系統也很重要。下列段落會簡要地介紹主要功能，以及它們在APM 及ACPI 電源管理系統中的用途：

待命 此操作模式關閉畫面顯示。在某些電腦上，會調節(Throttling) 處理器的效能。並不是所有的APM 實行都具有此功能。此功能等同於ACPI 狀態S1 或S2。

暫停(於記憶體) 此模式會將整個系統狀態寫入RAM 中。接著，除了RAM 以外，整個系統都會進入睡眠狀態。在此狀態中，電腦所使用的電源極少。此狀態的好處是在幾秒內將工作復原到暫停之前的狀態，而不用開機或重新啓動應用程式。使用APM 的設備通常會在蓋子關閉時暫停，而在打開蓋子時啓動。此功能等同於ACPI 狀態S3。對此狀態的支援仍在開發中，因此支援程度多半因硬體而異。

休眠(暫停寫入到磁碟) 在此操作模式，會將整個系統狀態寫入硬碟，然後關閉系統。要從此狀況重新啓動需耗時30 到90 秒。還原時會回到暫停前的狀態。有些製造商會為此模式提供有用的混合功能(例是IBM Thinkpad 中的RediSafe)。等同的ACPI 狀態為S4。在Linux 中，暫停寫入到磁碟是由獨立於APM 與ACPI 之外的核心常式來執行。

電池監視器 ACPI 與 APM 能檢查電池的變化狀態，並提供相關資訊。此外，兩個系統都會在電力到達某個關鍵狀態時，協調要執行的動作。

自動關閉電源 關機後，電腦會關閉電源。此功能很重要，尤其是在電池用盡前所執行的自動關機。

關閉系統元件 就整個系統而言，關閉硬碟是最省電的方式。依整個系統的穩定度而異，有時可讓硬碟進入休眠狀態。不過，睡眠時期的持續時間愈長，遺失資料的風險愈大。其他元件可透過 ACPI 停用(至少在理論上)，或是在 BIOS 設定中永久停用。

處理器速度控制 與 CPU 有關的三種省電方式如下：頻率和電壓比例(也稱為 PowerNow! 或 Speedstep)、調節(Throttling) 以及讓處理器進入睡眠(C 狀態)。依據電腦的操作模式，也可以合併這些操作方法。

16.2 APM

APM BIOS 本身能執行一些省電功能。許多筆記型電腦則可透過組合鍵或關閉蓋子等方式來啟動待命狀態或暫停狀態，而不需任何特別地操作系統功能。不過，如果要以指令啟動這些模式，必須先觸發一些動作才能使系統暫停。如果要檢視電池充電程度，需要有特別的程式套件與適用的核心。

SUSE LINUX 核心內建即支援 APM。不過，APM 只在 ACPI 未在 BIOS 中實行以及偵測到 APM BIOS 時才會啟動。如果要啟動 APM 支援，必須在開機提示中輸入 `acpi=off` 以關閉 ACPI。請輸入 `cat /proc/apm` 以檢查 APM 是否啟動。輸出中含有各種數字的話，表示一切正常。現在若使用 `shutdown-h` 指令，應可關閉電腦。

BIOS 實行若未能完全符合標準則會導致 APM 發生問題。有些問題可以使用特定的開機參數來加以避免。所有參數是以 `apm=parameter` 的形式輸入於開機提示中：

on or off 開啓或關閉 APM 支援。

(no)allow-ints 允許在 BIOS 功能執行期間出現中斷。

(no)-broken-psr BIOS 的“GetPowerStatus”功能無法正常運作。

(no)-realmode-power-off 在關機前，重新設定處理器至真實模式。

(no)-debug 在系統記錄中記錄 APM 事件。

(no-)power-off 在關機後關閉系統電源。

bounce-interval=*n* 暫停事件發生後，要經過多少時間才能執行下個暫停事件，該時間以0.01秒為單位。

idle-threshold=*n* 執行BIOS的idle功能後，系統閒置的百分比(0=永遠，100=永不)。

idle-period=*n* 系統啟動後所經過的時間，該時間以0.01秒為單位。

APM精靈(apmd)已停用。其功能目前是由新的powersaved處理，且能支援ACPI和CPU頻率比例。

16.3 ACPI

ACPI(進階組態與電源介面)可讓作業系統設定和控制個別的硬體元件。ACPI可取代PnP與APM。它能提供一些資訊，包括電池、變電器、溫度、風扇以及“關閉蓋子”或“電池電力不足”等系統事件。

BIOS會提供一些表格，內含關於個別元件與硬體的存取方法等資訊。作業系統會使用這些資訊來執行工作，像是指定中斷或啟動和關閉元件。因為作業系統會執行儲存於BIOS中的指令，所以BIOS實行會決定其功能。ACPI能偵測和載入的表格在/var/log/boot.msg中可以找到。請參閱節16.3.4,“疑難排解”，以取得更多有關ACPI問題疑難排解的資訊。

16.3.1 ACPI的動作

如果核心在系統啟動時偵測到ACPI BIOS，會自動啟動ACPI並關閉APM。有些較舊的機器可能需要用到開機參數acpi=on。電腦需支援ACPI 2.0或以後的版本。請檢查/var/log/boot.msg中的核心開機訊息，以查看ACPI是否啟動。接著，需載入一些模組。這會由ACPI精靈的啟動程序檔來完成。如果其中任何一個模組導致問題發生，該項模組就不會在/etc/sysconfig/powersave/common中載入或取消載入。系統記錄(/var/log/messages)內有模組的訊息，從中可以知道已偵測到哪些元件。

/proc/acpi目前含有一些檔案，用來提供有關系統狀態的資訊，也可以用來對某些狀態進行變更。有些功能還不能使用，因為仍在開發中，而且有些功能的支援主要是依靠製造商是否在產品中實行。

所有檔案(dsdt與fadt除外)都可使用cat來讀取。有些檔案可使用echo來修改設定，例如，echo X > file可為X指定適合的值。此外，務必使用powersave指令來存取此資訊以及控制選項。最重要的檔案說明如下：

- `/proc/acpi/info` 有關ACPI的一般資訊。
- `/proc/acpi/alarm` 在此指定何時從睡眠狀態中喚醒系統。目前，尚未完整支援此功能。
- `/proc/acpi/sleep` 提供可能的睡眠狀態的相關資訊。
- `/proc/acpi/event` 所有事件都會會在此報告，並經由Powersave精靈來加以處理(powersaved)。如果沒有精靈存取此檔案，快速按一下(電源)按鈕或是關閉蓋子等事件，可使用`cat/proc/acpi/event`來讀取(按Ctrl-C來終止)。
- `/proc/acpi/dsdt` 與 `/proc/acpi/fadt`
這些檔案包含ACPI的(不同系統說明表格) DSDT表格和(固定ACPI說明表格) FADT表格。可以使用`acpidmp`、`acpidisasm`與`dmdecode`來讀取它們。這些程式及其文件位於`pmtools`套件中。例如，`acpidmpDSDT | acpidisasm`。
- `/proc/acpi/ac_adapter/AC/state`
顯示是否已連接AC轉換器。
- `/proc/acpi/battery/BAT*/{E}` 有關電池狀態的詳細資訊。透過比較資訊中的上次完整電量以及狀態中的剩餘電量來讀取充電等級。另一種更方便的方法，便是使用在節16.3.3, "ACPI工具"中所介紹的特別程式。可在提醒中指定觸發電池事件的充電等級。
- `/proc/acpi/button` 此目錄包含各種切換的資訊。
- `/proc/acpi/fan/FAN/state` 顯示風扇是否正在運作。以手動方式透過在此檔案中寫入0(開啓)或3(關閉)，以開啓或關閉風扇。不過，在過熱時，核心與硬體(或BIOS)中的ACPI程式碼都會覆寫此設定。
- `/proc/acpi/processor/CPU*/info`
關於處理器的省電選項的資訊。
- `/proc/acpi/processor/CPU*/power`
關於目前處理器狀態的資訊。在C2旁邊有星號表示處理器閒置中。查看使用值時，這是最常出現的狀態。
- `/proc/acpi/processor/CPU*/throttling`
能用來設定調節處理器的時脈。通常，調節可以有八個層級。這和CPU的頻率控制是兩回事。

`/proc/acpi/processor/CPU*/limit`

如果是由精靈來自動控制效能(過時)及調節功能，則可在此指定最大上限。部份限制是由系統所決定。部份則可由使用者來調整。

`/proc/acpi/thermal_zone/` 每個溫度區都有一個子目錄。溫度區是指一個具有類似的溫度屬性的區域，由硬體製造商指定區域的數值及名稱。不過大部份由ACPI所提供的可能選項，很少被實作。通常反而是使用BIOS來控制溫度。作業系統很少有干預的機會，因為這可能會減少硬體的使用壽命。因此，以下有一部份的說明僅具理論價值。

`/proc/acpi/thermal_zone/*/temperature`

溫度區目前的溫度。

`/proc/acpi/thermal_zone/*/state`

指示是否一切都正常，或ACPI是否套用主動或被動冷卻規則。如果風扇控制獨立於ACPI之外，則狀態會一直是正常。

`/proc/acpi/thermal_zone/*/cooling_mode`

選取由ACPI控制的冷卻方法。從被動(效能較低，經濟型)選擇或是作用中的冷卻模型(完整的效能、風扇噪音)。

`/proc/acpi/thermal_zone/*/trip_points`

啟用判定溫度上限以觸發指定動作的功能，像是被動或主動冷卻、暫停(過熱)或關機(嚴重)。在DSDT(視設備而有所不同)中定義可能動作。在ACPI指定中的啓動點分別是嚴重、過熱、被動、主動1及主動2。即使不會每個都會實作，仍必須依此順序將它們輸入在此檔案中。例如，`echo 90:0:70:0:0 > trip_points` 項目即是將嚴重的溫度設為90，將被動設為70(以攝氏測量的所有溫度)。

`/proc/acpi/thermal_zone/*/polling_frequency`

如果在temperature檔案中的值，未在溫度變更時自動更新，請在此切換輪詢模式。`echoX > /proc/acpi/thermal_zone/*/polling-frequency` 指令能限定每X秒查詢一次溫度。設定X=0以關閉輪詢。

這些設定、資訊及事件都不需以手動方式編輯。可以使用Powersave精靈(powersaved)和各種應用程式來完成，像是powersave、kpowersave及wmpowersave。請參閱節16.3.3, "ACPI 工具"。因為powersaved包含較舊的acpid功能，所以已不再需要acpid。

16.3.2 控制CPU 效能

CPU 有三種省電方法。依據電腦的操作模式的不同，這些方法可合併使用。省電也表示能降低系統溫度，減低風扇的使用頻率。

頻率和電壓比例 AMD 與Intel 分別以PowerNow! 與Speedstep 來實作此技術。不過，此技術也套用於其他製造商的處理器中。CPU 的時脈頻率及其核心電壓會同時降低，產生高於線性的省電效能。也就是當頻率減半(效能減半)，使用的電量卻能遠低於原本的一半。此技術與APM 或ACPI 無關，需使用精靈來調整適用的頻率及目前所需的效能。您可以在 `/sys/devices/system/cpu/cpu*/cpufreq/` 目錄中進行設定。

調節時脈頻率 此技術會忽略部分CPU 的時脈訊號脈衝。到達25% 調節時，會省略四分之一脈衝，到達87.5% 時則每八次脈衝僅有一次會到達處理器。不過，節省用電量稍低於線性。通常調節功能僅在無此頻率比例時使用，或是為了最大化省電效果時使用。此外，此技術必須使用特定程序來進行控制。系統介面是 `/proc/acpi/processor/*/throttling`。

使處理器進入睡眠 作業系統會在沒事可做時使處理器進入睡眠。在此情況中，作業系統會傳送halt 指令給CPU。共有三種狀態：C1、C2 及C3。在最節省的C3 狀態中，連處理器快取與主記憶體間的同步化也會暫停。因此，僅能在沒有任何設備透過Bus master 活動來修改主記憶體內容時可以套用此狀態。有些驅動程式會禁止使用C3。目前的狀態會顯示在 `/proc/acpi/processor/*/power` 中。

頻率比例及調節只在處理器忙碌時使用，因為在處理器閒置時，一定會套用最節省的C 狀態。如果CPU 正忙碌，頻率比例是建議的省電方法。通常處理器僅有部份的工作負載。在此情況中，可以使用較低的頻率。通常會使用精靈來控制動態頻率比例，例如powersaved 是最佳方法。對電池的操作而言，靜態設定為較低頻率比較好，也可以用在您想降低電腦溫度或減低噪音時。

調節應做最後手段使用，例如，在高度系統負載下仍要延伸電池操作時間時。不過在調節過多時，有些系統無法運作順暢。此外，當CPU 要做的事不多時，調節CPU 是無意義的動作。

在SUSE LINUX 中，這些技術是由Powersave 精靈所控制。組態的說明位於節16.5, "powersave 套件"。

16.3.3 ACPI 工具

ACPI 公用程式包含僅顯示電池充電等級與溫度等資訊的工具(acpi、klaptopdaemon 及wmacpimon 等等。)、協助在 `/proc/acpi` 中存

取架構或協助監控變更(akpi、acpiw、gtkacpiw)的工具，以及在BIOS中編輯ACPI表格的工具(pmtools套件)。

16.3.4 疑難排解

共有兩種不同類型的問題。一方面是核心的ACPI程式碼包含無法及時偵測到的錯誤。在這種情況中，將會有可供下載的解決方案。不過通常問題是因BIOS而起。有時，會刻意在BIOS中整合與ACPI規格不符的技術，以避免在其他常見作業系統中的ACPI實作錯誤。會在黑名單中將那些在ACPI實行中有重大錯誤的硬體元件記錄下來，以避免Linux核心對這些元件使用ACPI。

發生問題時要做的第一件事是更新BIOS。如果電腦未能開機，下列中的某一個開機參數也許有幫助：

`pci=noacpi` 不使用ACPI來設定PCI設備。

`acpi=oldboot` 僅執行一個簡單的資源組態。不將ACPI用於其他目的。

`acpi=off` 關閉ACPI。

┌

未使用ACPI的開機問題

有些較新的機器(尤其是SMP系統及AMD64系統)需透過ACPI以正確設定硬體。關閉這些機器的ACPI會發生隨之而來的問題。

└

開機後，可使用`dmesg | grep -2i acpi`指令來監控系統的開機訊息(或所有訊息，因為也可能是ACPI以外的因素構成問題)。如果是在分析ACPI表格時發生問題，則最重要的—the DSDT—表格，可用改良版本來替換。在此情況中，會忽略BIOS的錯誤DSDT。程序在節16.5.4,“疑難排解”中描述。

在核心組態中，有個啓動ACPI除錯訊息的切換。如果已編譯並安裝好一個具有ACPI除錯能力的核心，則專家將能取得詳細資訊支援，以便搜尋錯誤。

如果您曾遇到BIOS問題或硬體問題，建議您聯絡製造商。尤其是哪些一直未提供Linux支援的製造商，更應該出面解決這些問題。唯有讓製造商得知他們有不少使用Linux的客戶，他們才會嚴肅地處理這些問題。

若需詳細資訊

ACPI 的其他文件和說明：

- <http://www.cpqlinux.com/acpi-howto.html> (詳細的ACPI HOWTO，內含DSDT 修補程式)
- <http://www.intel.com/technology/iapc/acpi/faq.htm> (ACPI FAQ @Intel)
- <http://acpi.sourceforge.net/> (Sourceforge 的ACPI4Linux 計劃)
- <http://www.poupinou.org/acpi/> (Bruno Ducrot 的DSDT 修補程式)

16.4 硬碟的休眠

在Linux 中，可在不需使用硬碟時，讓硬碟完全進入睡眠狀態，或是讓硬碟以更省電、更安靜的方式來運作。在目前的筆記型電腦中，您不用手動關閉硬碟，因為它們會在不用的時候自動進入省電操作模式。不過，如果您想最大化省電效果，可嘗試下列幾種方法。powersaved 能控制大部份這方面的功能。

hdparm 應用程式能修改各種硬碟設定。-y 選項能立即將硬碟切換到待命模式。-Y (警告) 能讓她進入睡眠。hdparm -S x 則會讓硬碟閒置一段時期後關閉。可將(x) 換成以下內容：0 能關閉此機制，讓硬碟持續運作。1 到240 的值將乘以五秒。241 到251 的值則是以30 分鐘為一個單位，依序從30 分鐘的閒置到11 倍的330 分鐘的閒置。

可以使用-B 選項來控制硬碟內部的省電選項。可從0 到255 中選取一個值，以最大化省電效果或最大化電力輸出。其結果視硬碟用途而定，難以評估。如果要減少硬碟噪音，請使用-M 選項。從128 到254 中選取一個值，以決定要安靜或快速。

通常，要讓硬碟進入睡眠不是件容易的事。在Linux 中，會有多個程序寫入硬碟中，因而重複喚醒硬碟。因此，有必要去瞭解Linux 如何處理那些要寫入硬碟的資料首先，會將所有資料在RAM 中做緩衝處理。核心更新精靈(kupdated) 可以監控緩衝區。當資料到達特定的時間限制，或當緩衝區已填滿至某一程度時，會將緩衝區的內容注入硬碟。緩衝區的大小則動態地由記憶體地的大小及系統負載來決定。根據預設，kupdated 會設成較短的間隔，以最大化資料的完整性。它會每5 秒檢查一次緩衝區，並會在資料時間大於30 秒或緩衝區的容量已達30% 時通知bdflush。接著，bdflush 精靈會將資料寫入硬碟。它也可以在不經由kupdated 之下寫入硬碟，例如，在緩衝區已滿時。



損害資料完整性

變更核心更新精靈的設定有害資料的完整性。



除了這些程序之外，像是ReiserFS與Ext3等日誌檔案系統，不經由bdflush會將中繼資料寫入硬碟，也會使得硬碟無法停止運作。為了避免這類情形，正在開發適用於行動設備的核心延伸程式。請參閱`/usr/src/linux/Documentation/laptop-mode.txt`以取得詳細資訊。

另一個重要因素在於啓動程式的行為方式。例如，好的編輯器會定期為修改中的檔案，將隱藏備份檔寫入硬碟，因而喚醒硬碟。停用這類功能可能會傷害資料的完整性。

與此相關，postfix郵件精靈會使用POSTFIX_LAPTOP變數。如果將此變數設為yes，postfix會減少存取硬碟的頻率。不過，若增加kupdated的時間間隔，則此項設定將無關緊要。

16.5 powersave 套件

powersave 套件負責膝上型電腦在電池操作時的省電功能。它的部份功能對於一般工作站和伺服器也很有用，例如暫停、待命、ACPI 按鈕功能和使IDE 硬碟休眠等。

此套件包含電腦的所有電源管理功能。它支援使用ACPI、APM、IDE 硬碟和PowerNow! 或SpeedStep 技術的硬體。來自套件apmd、acpid、ospmid 和cpufreqd (現在為cpuspeed) 的功能已合併於powersave 套件中。來自這些套件的精靈應該無法與powersave 精靈同時執行。

即使您的系統不包含上述所有硬體元件，仍可以使用powersave 精靈來控制省電功能。因為ACPI 和APM 互斥，僅可在電腦上使用其中一種系統。精靈會自動偵測硬體組態的變化。

關於powersave 的資訊

關於powersave 套件的資訊也可以在`/usr/share/doc/packages/powersave` 中找到。

16.5.1 設定powersave 套件

一般而言，powersave 的組態散佈於數個檔案：

`/etc/sysconfig/powersave/common`

此檔案包含powersave精靈的一般設定。例如，可以增加變數**POWERSAVE_DEBUG** 的值以增加偵錯訊息(在/var/log/messages 中)的數量。

`/etc/sysconfig/powersave/events`

powersave 精靈需要此檔案以處理系統事件。事件可以是指定的外部動作或精靈本身執行的動作。對於外部動作，精靈會嘗試執行/usr/lib/powersave/scripts/ 中的執行檔。預先定義內部動作：

- ignore
- throttle
- dethrottle
- suspend_to_disk
- suspend_to_ram
- standby
- do_suspend_to_disk
- do_suspend_to_ram
- do_standby

throttle 以**POWERSAVE_MAX_THROTTLING** 中定義的值減緩處理器。此值視目前的配置而定。**dethrottle** 將處理器設為完整效能。**suspend_to_disk**、**suspend_to_ram** 和**standby** 會觸發休眠模式的系統事件。這三個動作一般負責觸發休眠模式，但是它們應該永遠與特定系統事件關聯。

目錄/usr/lib/powersave/scripts 包含處理事件的程序檔：

notify 藉由主控台、X 視窗或柔和式聲響訊號來通知事件。

screen_saver 啟用螢幕保護裝置。

switch_vt 在暫停或待命之後，如果螢幕發生錯置時會很有用。

wm_logout 儲存設定並從GNOME、KDE 或其他視窗管理員登出。

wm_shutdown 儲存GNOME 或KDE 設定並關閉系統。

例如，如果設定 `POWERSAVE_EVENT_GLOBAL--SUSPEND2DISK="prepare_suspend_to_disk do_suspend_to_disk"` 變數，兩個程序檔或動作會在使用者給 `powersave` 休眠模式指令 `suspend to disk` 時，立即以特定順序處理。精靈會執行外部程序檔 `/usr/lib/powersave/scripts/prepare_suspend_to_disk`。在此程序檔成功處理之後，精靈會執行內部動作 `do_suspend_to_disk`，並在程序檔卸載關鍵模組並停止服務之後，將電腦設為休眠模式。

`(sleep)` 按鈕事件的動作可在 `POWERSAVE_EVENT_BUTTON_SLEEP="notify_suspend_to_disk"` 中修改。在此例中，外部程序檔 `notify` 會通知使用者暫停。之後，會產生 `POWERSAVE_EVENT_GLOBAL.SUSPEND2DISK` 事件，導致執行上述動作及安全系統暫停模組。可使用 `/etc/sysconfig/powersave/common` 中的變數 `POWERSAVE_NOTIFY_METHOD` 自訂程序檔 `notify`。

`/etc/sysconfig/powersave/cpufreq`
包含最佳化動態 CPU 頻率設定的變數。

`/etc/sysconfig/powersave/battery`
包含電池限制和其他電池特定設定。

`/etc/sysconfig/powersave/sleep`
在此檔案中，啓用休眠模式並決定應卸載哪個關鍵模組，以及在暫停或待命事件之前應停止的服務。當系統繼續時，這些模組會重新載入並啓動服務。您甚至可以延遲觸發的休眠模式(以便儲存檔案)。預設值主要考量 USB 和 PCMCIA 模組。特定模組通常會造成暫停或待命的錯誤。請參閱節 16.5.4, "疑難排解" 以取得關於辨識錯誤的詳細資訊。

`/etc/sysconfig/powersave/thermal`
啓用冷卻和熱控制。關於此主題的詳細資訊也可在檔案 `/usr/share/doc/packages/powersave/README.thermal` 中找到。

`/etc/sysconfig/powersave/scheme_*`
這些為搭配特定部署狀況電源消耗的不同配置。數種配置已預先設定好並可供使用。可在此儲存自訂配置。

16.5.2 設定 APM 和 ACPI

暫停和待命

根據預設，休眠模式未啓用，因為它們在部份電腦上没有作用。有三種基本 ACPI 休眠模式和兩種 APM 休眠模式：

Suspend to Disk (ACPI S4、APM 暫停)

將全部記憶體內容儲存到硬碟。電腦完全關閉，不消耗任何電力。

Suspend to RAM (ACPI S3、APM 暫停)

將所有設備的狀態儲存到主要記憶體。僅主要記憶體繼續消耗電力。

Standby (ACPI S1、APM 待命) 關閉部份設備(視製造商而定)。

請確定檔案 `/etc/sysconfig/powersave/events` 中的下列預設選項設為暫停、待命和繼續的正確處理(預設值為在安裝SUSE LINUX 之後)：

```
POWERSAVE_EVENT_GLOBAL_SUSPEND2DISK=
    "prepare_suspend_to_disk do_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_SUSPEND2RAM=
    "prepare_suspend_to_ram do_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_STANDBY=
    "prepare_standby do_standby"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2DISK=
    "restore_after_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2RAM=
    "restore_after_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_RESUME_STANDBY=
    "restore_after_standby"
```

自訂電池狀態

在檔案 `/etc/sysconfig/powersave/battery` 中，定義三種電池充電強度(以百分比)，當達到時會觸發警示或特定動作。

```
POWERSAVED_BATTERY_WARNING=20
POWERSAVED_BATTERY_LOW=10
POWERSAVED_BATTERY_CRITICAL=5
```

當充電強度掉到特定限制之下，要執行的動作或程序檔定義於組態檔 `/etc/sysconfig/powersave/events` 中。按鈕的標準動作可以修改為如節16.5.1, "設定powersave 套件" 中所述。

```
POWERSAVE_EVENT_BATTERY_NORMAL="ignore"
POWERSAVE_EVENT_BATTERY_WARNING="notify"
POWERSAVE_EVENT_BATTERY_LOW="notify"
POWERSAVE_EVENT_BATTERY_CRITICAL="wm_shutdown"
```

將電源消耗與不同條件搭配

系統行為可與電源供應類型搭配。當系統從AC 電源供應中斷連線並以電池操作時，系統的電源消耗應減少。相似地，系統一連線到AC 電源供應時，效能應自動增加。CPU 頻率、IDE 的省電功能和許多其他參數均可修改。

當電腦連線到或從AC 電源供應中斷連線時，要執行的動作定義於/etc/sysconfig/powersave/events 中。請在/etc/sysconfig/powersave/common 中選取要使用的配置：

```
POWERSAVE_AC_SCHEME="performance"  
POWERSAVE_BATTERY_SCHEME="powersave"
```

配置儲存在/etc/sysconfig/powersave 中的檔案內。檔案名稱的格式為scheme_nameofthescheme。範例參考兩個配置：scheme_performance 和scheme_powersave。performance、powersave、presentation 和acoustic 為預先設定。藉由YaST 電源管理模組(請參閱節16.6, "YaST 電源管理模組") 的協助，可編輯、建立、刪除現有配置或和不同電源供應狀態相關聯。

16.5.3 其他ACPI 功能

如果您使用ACPI，可以控制系統對ACPI 按鈕(電源、休眠、開蓋、關蓋)的回應。請在/etc/sysconfig/powersave/events 中設定動作的執行。請參考此組態檔以取得個別選項的說明。

POWERSAVE_EVENT_BUTTON_POWER="wm_shutdown"

當按下電源按鈕時，系統回應為關閉對應視窗管理員(KDE、GNOME、fvwm 等)。

POWERSAVE_EVENT_BUTTON_SLEEP="suspend_to_disk"

當按下休眠按鈕時，系統設為休眠(suspend-to-disk) 模式。

POWERSAVE_EVENT_BUTTON_LID_OPEN="ignore"

當開啓蓋子時，不發生任何事。

POWERSAVE_EVENT_BUTTON_LID_CLOSED="screen_saver"

當關上蓋子時，會啓動螢幕保護裝置。

如果CPU 負載在特定時間未超過特定限制，則可以進一步節流CPU 效能。在POWERSAVED_CPU_LOW_LIMIT 中指定負載限制，在POWERSAVED_CPU_IDLE_TIMEOUT 中指定逾時。

16.5.4 疑難排解

所有錯誤訊息和警示會記錄在檔案 `/var/log/messages` 中。如果您找不到所需資訊，請使用 `/etc/sysconfig/powersave/common` 檔案中的 `DEBUG` 提高 `powersave` 訊息的資料詳細程度。將變數值增加到 7 或甚至 15，並重新啓動精靈。`/var/log/messages` 中更詳細的錯誤訊息可協助您找出錯誤。下列小節涵蓋 `powersave` 最常見的問題。

以硬體支援啓動 **ACPI**，但沒有作用

如果您遭遇到 **ACPI** 的問題，請使用下列指令來搜尋 **ACPI** 特定訊息的 `dmesg` 輸出：`dmesg|grep-i acpi`。必須更新 BIOS 以解決問題。請到您膝上型電腦製造商的首頁，尋找更新的 BIOS 版本並安裝。請詢問製造商以符合最新 **ACPI** 規格。在 BIOS 更新後，如果錯誤持續發生，請繼續以更新的 **DSDT** 取代您 BIOS 中的錯誤 **DSDT** 表格：

1. 從 <http://acpi.sourceforge.net/dsdt/tables> 下載您系統的 **DSDT**。檢查檔案是否已解壓縮，並以所示副檔名 `.aml` (**ACPI** 機器語言) 編譯。如果是此狀況，請繼續步驟 3。
2. 如果下載的表格副檔名為 `.asl` (**ACPI** 原始語言)，則必須以 `iasl` (`pmtools` 套件) 加以編譯。若要這樣做，請輸入 `iasl -sa file.asl` 指令。最新版本的 `iasl` (**Intel ACPI** 編譯器) 可在 <http://developer.intel.com/technology/iapc/acpi/downloads.htm> 中找到。
3. 將檔案 `DSDT.aml` 複製到任何位置(建議 `/etc/DSDT.aml`)。編輯 `/etc/sysconfig/kernel` 並將路徑與 **DSDT** 檔案搭配。啓動 `mkinitrd` (套件 `mkinitrd`)。只要您解除安裝核心並使用 `mkinitrd` 來建立 `initrd`，當系統啓動時，修改的 **DSDT** 便會整合並載入。

CPU 頻率沒有作用

請參考核心來源 (`kernel-source`) 以查看是否支援您的處理器。您需要特殊核心模組或模組選項以啓用 **CPU** 頻率控制。此資訊可在 `/usr/src/linux/Documentation/cpu-freq/*` 中找到。如果需要特殊核心模組或模組選項，可在檔案 `/etc/sysconfig/powersave/cpufreq` 中藉由變數 `CPUFREQD-MODULE` 和 `CPUFREQD-MODULE-OPTS` 加以設定。

暫停和待命沒有作用

有數種核心相關問題會使ACPI系統的暫停和待命無法使用：

- 目前超過1 GB RAM 的系統不支援暫停。
- 目前多處理器系統和P4 處理器(具有超執行緒)的系統不支援暫停。

錯誤也可能是導因於錯誤的DSDT 執行方式(BIOS)。如果是此狀況，請安裝新的DSDT。

在ACPI 和APM 系統上：當系統嘗試卸載錯誤模組時，系統會停止或不觸發暫停事件。如果您不卸載模組或停止服務(導致無法成功暫停)，也可能發生此狀況。在兩種情況下，都請嘗試辨識無法啟動休眠模式的錯誤模組。在此狀況下，在/var/log/sleepmode 中的powersave 精靈所產生的記錄檔非常有用。如果電腦無法進入休眠模式，原因在於最後卸載的模組。請操作/etc/sysconfig/powersave/sleep 中的下列設定，在暫停或待命之前卸載有問題的模組。

```
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND2DISK=""
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND2RAM=""
POWERSAVE_UNLOAD_MODULES_BEFORE_STANDBY=""
POWERSAVE_SUSPEND2DISK_RESTART_SERVICES=""
POWERSAVE_SUSPEND2RAM_RESTART_SERVICES=""
POWERSAVE_STANDBY_RESTART_SERVICES=""
```

如果您在變動網路環境中或以遠端裝載檔案系統(如Samba 和NIS) 的連線使用暫停或待命，請在上述變數中使用自動裝載器以裝載或新增對應服務，例如smbfs 或nfs。如果應用程式在暫停或待命之前存取遠端裝載檔案系統，則服務無法正確停止，而檔案系統無法正確取消裝載。在系統繼續之後，檔案系統可能損毀並必須重新裝載。

使用ACPI，而Powersave 未注意到電池限制

使用ACPI，作業系統可以要求BIOS 在電池充電強度掉到特定限制之下時傳送訊息。此方法的好處在於不需要持續輪詢電池狀態，因為這可能損害電腦的效能。但是，當充電強度掉到特定限制之下時，即使假設BIOS 支援此功能，仍可能不發生此通知。如果系統發生此狀況，請將檔案/etc/sysconfig/powersave/battery 中的變數POWERSAVED_FORCE_BATTERY_POLLING 設為yes 以強迫電池輪詢。

16.6 YaST電源管理模組

YaST 電源管理模組可以設定上述所有電源管理設定。在您以‘系統’→‘電源管理’從YaST「控制中心」啟動模組時，會開啓模組的第一個對話方塊。如圖形16.1, “配置選取區域”所示。



Figure 16.1: 配置選取區域

在此對話方塊中，選取電池操作和AC操作要使用的配置。若要新增或修改配置，請按一下‘編輯配置’，便可開啓如圖形16.2, “現有配置的綜覽”所示的現有配置綜覽。

在配置綜覽中，選取要修改的配置，再按一下‘編輯’。若要建立新的配置，請按一下‘新增’。兩種狀況開啓的對話方塊均相同，如圖形16.3, “新增配置”中所示。

首先，為新的或編輯的配置輸入適合的名稱和描述。判斷此配置是否應控制CPU效能及如何控制。決定是否應使用頻率比例和調節設定以及使用到什麼程度。在硬碟的下列對話方塊中，定義最大效能或省電的‘待機規則’。‘柔和式聲響規則’控制硬碟的噪音強度(少數硬碟支援)。**‘冷卻規則’**決定要使用的冷卻方法。不幸地，BIOS很少支援此類型的熱控制。請讀取 `/usr/share/doc/packages/powersave/README.thermal` 以瞭解如何

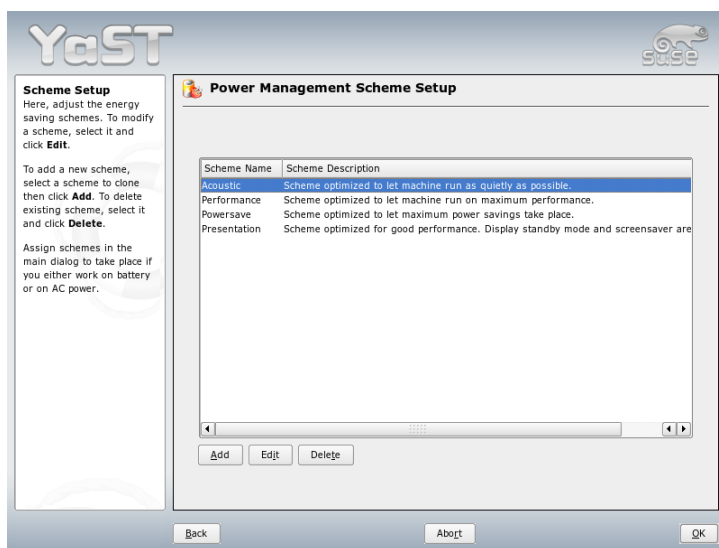


Figure 16.2: 現有配置的綜覽

使用風扇和被動冷卻方法。按一下‘下一步’以繼續設定連線的顯示器省電模式的對話方塊。選取‘啓用螢幕保護裝置’核取方塊以減少當電腦不使用時顯示器的電源消耗。‘啓用顯示器電源管理’可讓您控制顯示器在多久時間之後應進入待命、暫停或關閉電源模式。完成所有配置設定並按一下‘確定’以回到開始對話方塊。在開始對話方塊中，將自訂配置指定為兩種操作模式的其中一種。若要啓用您的設定，請以‘確定’結束此對話方塊。

也可使用‘電池警告’、‘ACPI 設定’或‘啓用暫停’從起始對話方塊進行全域電源管理設定。按一下‘電池警告’以存取電池充電強度對話方塊，如圖形16.4, “電池充電強度” 中所示。

當充電強度掉到特定可設定的限制之下時，您系統的BIOS會通知作業系統。在此對話方塊中，定義三個限制：‘警告電量’、‘低電量’和‘極低電量’。指定當充電強度掉到此限制之下時，要觸發的動作。通常，前兩種狀態僅觸發對使用者的通知。第三種重要強度會觸發關機，因為剩餘電力不足以繼續系統操作。選取適合的充電強度和想要的動作，然後按一下‘確定’以回到開始對話方塊。

使用‘ACPI 設定’存取設定ACPI按鈕的對話方塊。如圖形16.5, “ACPI 設定” 所示。ACPI 按鈕的設定決定系統對特定切換應如何回應。設定系統對按下電源按鈕、按下休眠按鈕和關閉膝上型電腦蓋子加以回應。按一下‘確定’以完成組態並

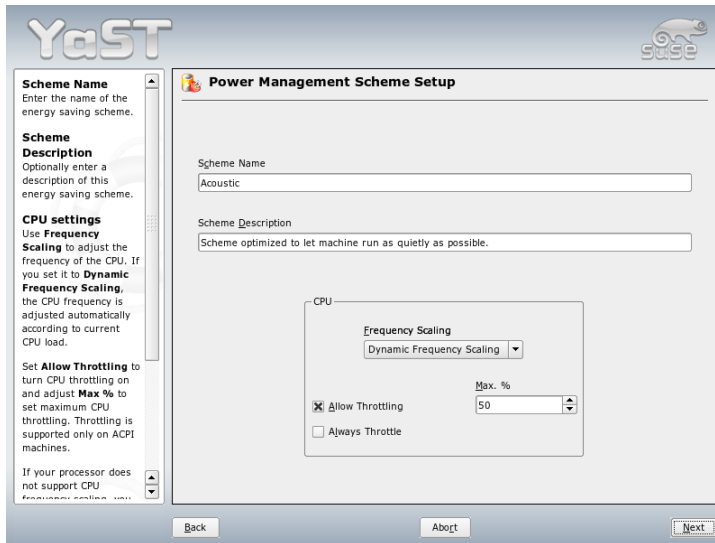


Figure 16.3: 新增配置

回到開始對話方塊。

按一下‘啟用暫停’以進入對話方塊，決定此系統使用者是否可使用暫停或待命功能，以及使用方式。按一下‘確定’以回到主要對話方塊。再按一下‘確定’以結束模組並確認您的電源管理設定。

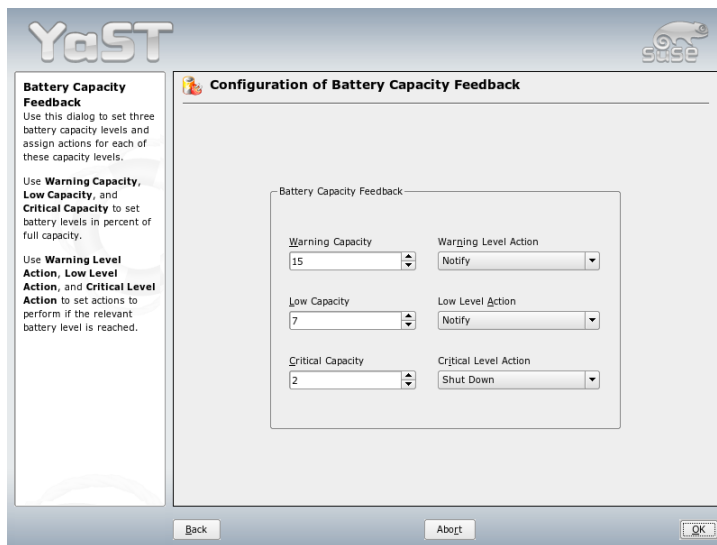


Figure 16.4: 電池充電強度

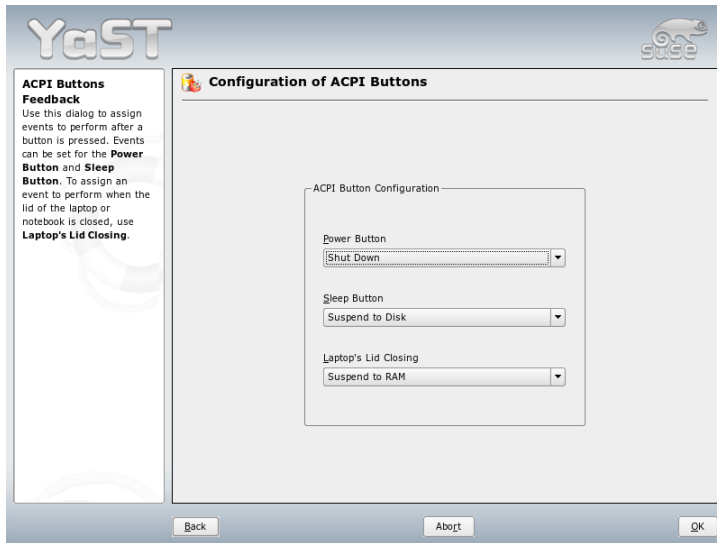


Figure 16.5: ACPI 設定

無線通訊

藉由使用Linux系統，您可以與其它電腦、手機或週邊設備進行各種通訊。WLAN(無線區域網路)可用來建立筆記型電腦網路。藍芽可用來連接個別的系统元件(滑鼠、鍵盤)、週邊設備、手機、PDA及個人電腦。IrDA大多用來與PDA或手機進行通訊。本章節將介紹這三種技術及其組態設定。

17.1	無線區域網路	274
17.2	藍芽	281
17.3	紅外線資料傳輸	289

17.1 無線區域網路

無線區域網路已成為行動電腦與網路不可或缺的一環。如今，大多數的筆記型電腦都有內建的WLAN卡。WLAN卡所使用的無線通訊802.11標準是由IEEE組織所制定。此標準最初用於最大傳輸率2 MBit/s。其間已增加許多新的標準來提高資料傳輸率。這些新標準對於類比調節、傳輸輸出及傳輸率等細節均有定義：

Table 17.1: WLAN 標準綜覽

名稱	頻段(GHz)	最大傳輸率(MBit/s)	附註
802.11	2.4	2	過時的；實際上無法取得終端設備
802.11b	2.4	11	普遍的
802.11a	5	54	較不普遍
802.11g	2.4	54	與11b 反向相容

此外還有一些專賣標準，像是德州儀器的802.11b變異標準，其最大傳輸率為22 MBit/s (有時稱為802.11b+)。不過使用此標準的網路卡數量有限。

17.1.1 硬體

SUSE LINUX 不支援802.11卡。支援大部份使用802.11a、802.11b和802.11g的卡。新卡通常符合802.11g標準，但也有一些是802.11b的卡。一般來說，支援附有下列晶片的卡：

- Lucent/Agere Hermes
- Intel PRO/Wireless 2100、2200BG 及2915ABG
- Intersil Prism2/2.5/3
- Intersil PrismGT
- Atheros 5210、5211 及5212
- Atmel at76c502、at76c503、at76c504 及at76c506
- Texas Instruments ACX100 及ACX111

也支援一些幾乎不再使用，並且無法取得的舊卡。「AbsoluteValue 系統」的網站上可找到詳細的WLAN卡清單及其使用的晶片種類：http://www.linux-wlan.org/docs/wlan_adapters.html.gz 及 <http://wiki.uni-konstanz.de/wiki/bin/view/Wireless/ListeChipsatz> 提供您各種WLAN晶片的綜覽。

某些網路卡需要在啓動驅動程式時載入韌體影像。例如Intersil PrismGT以及Atmel ACX100、ACX111。使用YaST線上更新，即可輕易地安裝韌體。Intel PRO-Wireless卡的韌體會與SUSE LINUX一起提供，並且會在偵測到此類型時，由YaST自動安裝。在裝好系統的/usr/share/doc/packages/wireless-tools/README.firmware可找到相關資訊。

您可以使用ndiswrapper應用程式來使用本來Linux不支援的卡。Ndiswrapper會使用與多數WLAN卡一同提供的Windows驅動程式。若需有關ndiswrapper的說明，請參閱/usr/share/doc/packages/ndiswrapper/README.SUSE(假定已安裝ndiswrapper套件時)。若需ndiswrapper的相關資訊，請參閱專案網站，網址為<http://ndiswrapper.sourceforge.net/support.html>。

17.1.2 功能

本章節包含無線網路的基本資訊。瞭解不同的作業模式、驗證和加密類型。

操作模式

基本上，無線網路可分為管理網路和臨機操作網路。受管理的網路具備的管理元件：存取點。在此模式(又稱為基礎結構模式)之下，網路上所有WLAN工作站的連線都會通過該存取點，此存取點亦可連線至乙太網路。臨機操作網路中沒有存取點。工作站之間直接進行通訊。臨機操作網路中的傳輸範圍及連線工作站的數目相當有限。因此，存取點的效率通常較高。WLAN卡甚至可做為存取點。大部份的網路卡均支援此項功能。

無線網路比有線網路更容易受到攔截和危害，因此各項標準均包含驗證和加密方式。在較早版本的IEEE 802.11標準中，可在WEP條款下找到這些項目的說明。然而，WEP已證實不夠安全(請參閱節17.1.5, "安全性")，WLAN業者(組成「Wi-Fi聯盟」)已定義一項新的安全標準，稱為WPA，用來提高WEP的安全性。更新的IEEE 802.11i標準(又稱為WPA2，WPA建立於802.11i草擬版本)包含WPA及其它驗證和加密方式。

驗證

管理網路使用各種不同的驗證機制，以確定只有獲得授權的工作站才可進行連線：

開放 開放的系統，即不需要驗證的系統。任何工作站均可加入網路。但可使用WEP 加密(請參閱節17.1.2, “加密”)。

共用金鑰(根據IEEE 802.11) 此程序使用WEP 金鑰進行驗證。不過，並不建議採用此程序，因為它使WEP 金鑰更容易受到攻擊。攻擊者只需要截聽工作站與存取點間的通訊達足夠的時間就行了。在驗證過程中，雙方交換相同的資訊，先是以加密的形式，然後是以未加密形式。因此，利用適當的工具即可重建金鑰。由於此方式是以WEP 金鑰來驗證和加密，因此無法提高網路的安全性。擁有正確WEP 金鑰的工作站可以驗證、加密及解密。缺乏金鑰的工作站則無法解密已收到的封包。因此，不論是否需要驗證其身份，該工作站均無法進行通訊。

WPA-PSK (根據IEEE 802.1x) WPA-PSK (PSK 即Pre-Shared Key (預先共用金鑰)的縮寫)的作用方式與共用金鑰程序相似。所有連線工作站及存取點都要有相同的金鑰。此金鑰長度為256 位元，且通常以通關密語的方式輸入。本系統不需要如WPA-EAP 一樣複雜的金鑰管理，並且更適合個人使用。因此，WPA-PSK 有時稱為WPA 「家用」。

WPA-EAP (根據IEEE 802.1x) WPA-EAP 實際上並非驗證系統，而是用來傳送驗證資訊的協定。WPA-EAP 用來保護企業中的無線網路。在私人網路中幾乎很少用到。因此，WPA-EAP 有時稱為WPA 「企業」。

加密

可使用各種不同的加密方式，防止未經授權者讀取無線網路中交換的資料封包，或進入網路：

WEP (定義於IEEE 802.11 中) 此標準使用RC4 加密演算法，最初的金鑰長度為40 位元，後來增加為104 位元。視24 位元的啓始向量是否包含其中而定，其長度通常為64 位元或128 位元。然而此標準具有某些弱點。此系統所產生的金鑰可能受到攻擊。儘管如此，使用WEP 仍然比完全不加密的網路來得好。

TKIP (定義於WPA/IEEE 802.11i 中) 此金鑰管理協定定義於WPA 標準中，使用與WEP 相同的加密演算法，其弱點則均已消除。每個資料封包都有一個新的金鑰，因此攻擊這些金鑰等於白費力氣。TKIP 與WPA-PSK 必須搭配使用。

CCMP (定義於IEEE 802.11i 中) CCMP 說明金鑰管理。通常與WPA-EAP 搭配使用，但也可配合WPA-PSK 使用。根據AES 的規定所進行的加密，比WEP 標準下的RC4 加密更安全。

17.1.3 使用YaST 進行設定

若要設定您的無線網路卡，先啟動YaST ‘網路卡’ 模組。在‘網路位址設定’ 中，選取設備類型‘無線’ 然後按一下‘下一步’ 。

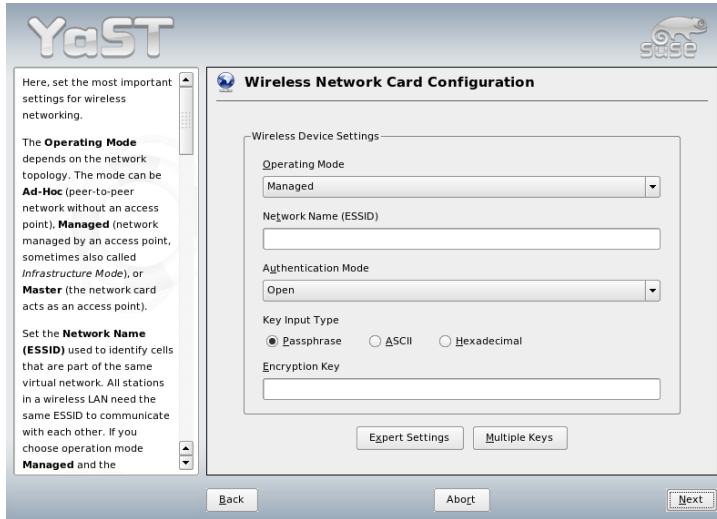


Figure 17.1: YaST：設定無線網路卡

在‘無線網路卡組態設定’ 中(顯示於圖形17.1, “YaST：設定無線網路卡” 中)，請指定WLAN 的基本操作設定：

操作模式 工作站可用三種模式來連接WLAN。依照要通訊的網路而定，適用不同的模式：‘隨機操作模式’(沒有存取點的對等網路)、‘管理模式’(由存取點管理網路) 或‘主要模式’(以您的網路卡為存取點)。

網路名稱(ESSID) 在無線網路中，所有工作站都要有相同的ESSID 才能互相通訊。在未指定的情況下，網路卡會自動選擇一個存取點，該存取點可能不是您想要使用的。

驗證模式 為您的網路選取適用的驗證方式：‘開放’、‘共用金鑰’ 或‘WPA-PSK’。如果您選取‘WPA-PSK’，則必須設定網路名稱。

進階設定 此按鈕會開啓一個對話方塊，說明WLAN 連線的組態設定細節。稍後會提供此對話方塊的詳細說明。

完成基本設定之後，您的工作站即可部署在WLAN 中。

無線網路的安全性

記得使用支援的驗證和加密方式，以保護您的網路資料傳輸。第三者可在未加密的WLAN 連線截取所有的網路資料。即使是不嚴密的加密(WEP) 也比不加密來得好。請參閱節17.1.2, “加密” 與節17.1.5, “安全性” 以取得更多資訊。

視所選取的驗證方式而定，YaST 提示會要求您在其它對話方塊中微調您的設定。不需要設定‘開放’ 的組態，因為它不需要驗證即可執行未加密的作業。

WEP 金鑰 設定金鑰輸入類型。選擇‘通關密語’、‘ASCII’ 或‘16 進位’ 其中之一。您最多可使用四個金鑰來加密傳送的資料。按一下‘多個金鑰’ 來開啓金鑰設定對話方塊。設定金鑰長度：‘128 bit’ 或‘64 bit’。預設值為‘128 位元’。在對話方塊底下的清單中，最多可指定四個金鑰，讓您的工作站用來加密。按‘設定為預設值’ 將其中之一設為預設金鑰。除非您對此做變更，否則YaST 都會使用第一個輸入金鑰為預設金鑰。如果標準金鑰被刪除了，則必須手動標記其它金鑰為預設金鑰。按一下‘編輯’ 來修改現有的清單項目或建立新金鑰。在此例中，快顯視窗會提示您選擇一個輸入類型(‘通關密語’、‘ASCII’ 或‘16 進位’)。如果您選取了‘通關密語’，請輸入一個單字或字元，會據此及先前指定的長度建立金鑰。‘ASCII’ 必須輸入5 個字元以建立64 位元金鑰；輸入13 個字元以建立128 位元金鑰。‘16 進位’ 則必須輸入10 個字元以建立64 位元金鑰，或26 個字元以在16 進位表示法中建立128 位元金鑰。

WPA-PSK 若要輸入一個WPA-PSK 金鑰，請選取‘通關密語’ 輸入法或‘16 進位’。在‘通關密語’ 模式下，必須輸入8 至63 個字元。在‘16 進位’ 模式下，必須輸入64 個字元。

按一下‘進階設定’ 以離開WLAN 連線的基本設定對話方塊，並進入進階組態。此對話方塊包含下列選項：

通道 應只有‘臨機操作’ 和‘主要’ 模式會用到WLAN 工作站的工作通道規格。在‘管理’ 模式下，網路卡會自動搜尋可用的通道以連接存取點。在‘臨機操作’ 模式中，必須從提供的12 個通道中選取其中之一，讓您的工作站可與其它工作站通訊。在‘主要’ 模式下，必須決定一個通道，讓您的網路卡可以用它來提供存取點功能。此選項的預設值為‘自動’。

位元率 視您網路的效能而定，您可以設定點對點間特定的傳輸位元率。在預設值為‘自動’的情況下，系統會試著選擇使用最高的資料傳輸率。有些WLAN卡不支援位元率的設定。

存取點 在擁有多個存取點的環境中，只要指定MAC位址即可預選其中一個存取點。

使用電源管理 當您出外時，可使用省電技術，使您的電池操作時間達到最久。如需電源管理的詳細資訊，請參閱章16, 電源管理。

17.1.4 公用程式

hostap (hostap 套件) 用來使WLAN卡發揮存取點的功能。如需更多有關此套件的資訊，請參閱專案首頁(<http://hostap.epitest.fi/>)。

kismet (kismet 套件) 是一種網路診斷工具，可用來傾聽WLAN的封包資料傳輸。您也可以藉此偵測您網路上任何侵入的企圖。如需更多相關資訊，請參閱<http://www.kismetwireless.net/> 及手冊。

17.1.5 設定WLAN的秘訣與技巧

瞭解如何調整WLAN的速度、穩定性及安全性。

穩定性及速度

無線網路的效能及可靠性，要看連線工作站是否能從其它工作站收到清楚的訊號。牆壁之類的障礙物會大大減弱訊號強度。訊號強度愈弱，傳輸速度愈慢。指令行(連結品質欄位)上的iwconfig公用程式或KDE中的kwifimanager可檢查運作時的訊號強度。若您的訊強度出現問題，試著將您的設備安裝在其它地方，或調整您存取點的天線方向。許多PCMCIA WLAN卡都有輔助天線，可大幅提高接收度。由廠商指定的速率(例如，54 MBit/s)為一額定值，代表推定的最大值。實際上，最大資料產生量還不到該值的一半。

安全性

如果您要建立一個無線網路，請記得在缺乏安全措施的情況下，任何在傳輸範圍內的人都可輕易地進入您的網路。因此，應確定啓用加密方式。所有WLAN卡和存取點都支援WEP加密。雖然不是安全無虞，但仍足以阻礙可能的攻擊。WEP通常足夠個人使用。WPA-PSK則是更佳的選擇，不過較舊型的存取點或路由器並未在其WLAN功能中使用PA-PSK。有些設備只要透過防火牆的

更新，即可執行WPA。此外，Linux並不支援所有硬體元件上的WPA。在開發此軟體時，WPA只適用於使用Atheros或Prism2/2.5/3晶片的網路卡。就後者而言，只有在使用hostap驅動程式的情況下，才能使用WPA（請參閱節17.1.6, “Prism2網路卡的問題”）。如果沒有WPA，則WEP仍然比完全不加密來得好。對於需要進階安全性的企業來說，只有在執行WPA的情況下才可操作無線網路。

17.1.6 疑難排解

如果您的WLAN卡沒有回應，檢查看看您是否已下載所需的韌體。請參照節17.1.1, “硬體”。以下段落為您說明一些可能出現的問題。

多重網路設備

現代的筆記型電腦通常具備一張網路卡和一張WLAN卡。如果您以DHCP來設定這兩者(自動指定位址)，則可能會出現名稱解析和預設閘道的問題。如果您可以偵測到路由器，卻無法瀏覽網際網路，表示已出現此問題。<http://portal.suse.com>的支援資料庫中有一篇與此相關的文章。如果要找這篇文章，請在搜尋對話方塊中輸入「DHCP」。

Prism2 網路卡的問題

有許多驅動程式可用於裝有Prism2晶片的設備。各種不同的網路卡多多少少都可以與不同的驅動程式配合運作。使用這些卡時，WPA只能配合hostap驅動程式來運作。如果這些網路卡無法順利運作，或完全無法運作，或者您想使用WPA，請參閱`/usr/share/doc/packages/wireless-tools/README.prism2`。

WPA

WPA支援已首度執行於SUSE LINUX中。Linux的WPA支援仍處於開發階段。因此，YaST只能設定WPA-PSK的組態。WPA無法用於許多網路卡。有些網路卡需要更新韌體，才能啟動WPA。如果您要使用WPA，請參閱`/usr/share/doc/packages/wireless-tools/README.wpa`。

17.1.7 若需詳細資訊

Linux「無線工具」的開發者Jean Tourrilhes在其網頁中提供大量有關無線網路的有用資訊。請參閱http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html。

17.2 藍芽

「藍芽」是連接各種設備(如：行動電話、PDA、週邊設備、膝上型電腦，或鍵盤或滑鼠等系統元件)的無線技術。此一名稱衍生自丹麥國王Harold Bluetooth，他統一了斯堪的那維亞許多分裂的地區。「藍芽」標誌是以“H”(像一顆星星)和“B”的北歐文字為基礎。

可由許多重要方面辨識「藍芽」和IrDA。首先，個別設備不需要直接「看到」彼此；其次，可將數個設備連線於一個網路中。但是，最大資料速率為720 Kbps(目前的版本1.2)。理論上，「藍芽」甚至可穿過牆面通訊。但是實際上，這要視牆面的材質和設備等級而定。有三種設備等級的傳輸範圍介於10至100公尺。

17.2.1 基本操作

下列小節簡述「藍芽」運作的基本原則。請瞭解需要符合哪些軟體需求，藍芽如何與系統互動，以及藍芽設定檔的運作方式。

軟體

若要使用「藍芽」，您需要「藍芽」介面卡(可以是內建介面卡或外部裝置)、驅動程式和「藍芽」通訊協定堆疊。Linux核心已經包含使用「藍芽」的基本驅動程式。Bluez系統做為通訊協定堆疊使用。為了確保應用程式可與「藍芽」搭配使用，必須安裝基礎套件bluez-libs和bluez-utils。這些套件提供許多所需的服務和公用程式。此外，部份介面卡(Broadcom、AVM BlueFritz!)要求必須安裝bluez-firmware套件。bluez-cups套件讓您可透過「藍芽」連線列印。

一般互動

「藍芽」系統由四個連鎖層所組成，提供需要的功能：

硬體 Linux核心支援所需的介面卡和適合的驅動程式。

組態檔 用於控制「藍芽」系統。

精靈 組態檔案所控制並可提供功能的服務。

應用程式 應用程式允許使用者使用並控制精靈所提供的功能。

當您插入「藍芽」介面卡時，熱插拔系統會載入對應的驅動程式。載入驅動程式之後，系統會檢查組態檔案以查看「藍芽」是否應該啓動。如果應該啓動，它會決定要啓動的服務。根據此資訊，會啓動對應的精靈。在安裝時會查探藍芽介面卡。如果找到一或多個介面卡，便會啓用藍芽。否則會停用藍芽系統。任何稍後新增的藍芽設備都必須以手動方式啓用。

設定檔

在「藍芽」中，服務是藉由設定檔(例如，檔案傳輸設定檔、基本列印設定檔和個人區域網路設定檔)所定義。若要讓設備使用其他設備的服務，兩者均必須了解相同設定檔—一些通常在設備套件上和手冊中遺失的資訊。不幸地，部份製造商並未嚴格符合個別設定檔的定義。即使如此，設備之間的通訊通常可順暢運作。

在下列文字中，本地設備為實體上連線到電腦的設備。其他僅可透過無線連線存取的設備稱為遠端設備。

17.2.2 組態

此小節介紹藍芽的組態。請瞭解需要哪些組態檔、需要哪些工具以及如何以YaST或手動方式設定藍芽。

以YaST設定藍芽

使用YaST「藍芽」模組(請參閱圖 圖形17.2, “YaST 藍芽組態”)來設定系統的「藍芽」支援。只要熱插拔偵測到系統上的「藍芽」介面卡(例如，在開機期間或是在插入介面卡時)，便會自動以此模組中設定的設定值來啓動「藍芽」。

在組態的第一步驟中，決定是否要在系統上啓動「藍芽」服務。如果您已啓用藍芽服務，可以設定兩個項目。首先，‘裝置名稱’。這是發現您的電腦時，其他設備所顯示的名稱。一共有兩種可能的保留字元—%h代表系統的主機名稱(有用，例如，如果它是由DHCP動態指定)，而%d會插入介面編號(如果您在電腦中有一個以上的「藍芽」介面卡才有用)。例如，如果您在欄位中輸入Laptop %h，而DHCP指定unit123的名稱給您的電腦，則其他遠端的設備將以Laptop unit123來識別您的電腦。

當遠端設備嘗試連線，第二個參數‘安全管理員’會與本機系統的行為相關。其差別在於處理PIN號碼。允許任何設備不需要PIN即可連線，或是判斷如何在需要PIN時選擇正確的PIN。您可以在適當的輸入欄位輸入PIN(儲存在組態檔案)。如果設備嘗試連線，它會先使用此PIN。如果它失敗，它就會恢復不使用PIN。就最佳的安全性而言，最好選擇第三個選項“永遠詢問使用者的PIN”。此選項允許您在不同(遠端)的設備上使用不同的PIN。

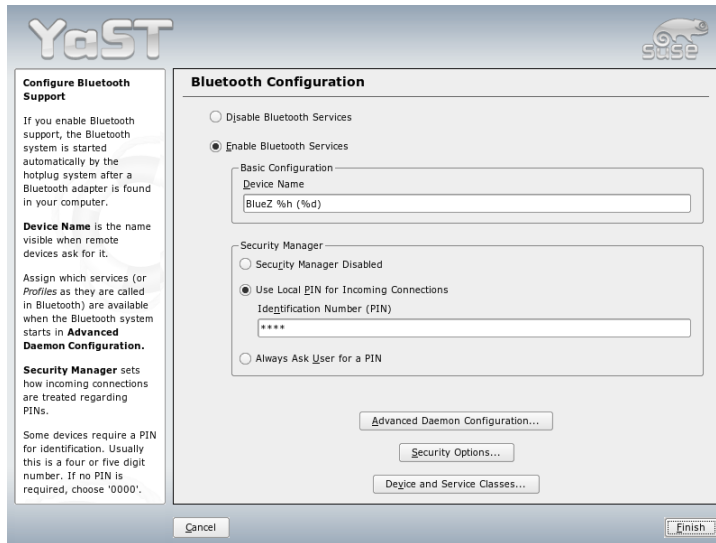


Figure 17.2: YaST 藍芽組態

然後按一下‘進階精靈組態’以進入選取和設定可用服務的對話方塊(在「藍芽」中也稱為「設定檔」)。所有可用的服務均顯示在清單中，可按一下‘啓用’或‘停用’加以啓用或停用。按一下‘編輯’以開啓快顯對話方塊，您可以在其中指定選取服務(精靈)的其他引數。除非您對服務很熟悉，否則請勿做任何變更。完成精靈的組態之後，按一下‘確定’以結束此對話方塊。

回到主要對話方塊，按一下‘安全性選項’以進入安全性對話方塊，您可以在其中指定加密、驗證和掃描設定。然後結束安全性對話方塊以回到主要對話方塊。以‘完成’關閉主要對話方塊之後，您的「藍芽」系統已經可以使用。

從主要對話方塊，也可以開啓‘設備和服務類別’對話方塊。藍芽設備群組成不同的「設備類別」。在此對話方塊中，針對電腦選擇正確的類別，例如「桌上型」或「膝上型」。設備類別與「服務類別」不同，並非很重要，同樣在此設定。有時像手機這一類的藍芽設備，如果它們可以偵測到您系統上所設定的正確服務類別，則只允許某些特定功能。這通常是發生在手機上，它們必須要有「物件轉送」的類別，才能從電腦轉送檔案或轉送檔案至電腦。您可以選擇多個類別。「爲了以防萬一」而選取所有的類別是沒有太大幫助的。預設的選擇應該在大部份的情況都適用。

如果您想要使用「藍芽」來設定網路，請在「進階精靈組態」對話方塊中啟用「PAND」，並以「編輯」設定精靈的模式。「藍芽」網路連線若要能運作，必須有一個pand在「傾聽」模式操作，而對等裝置則是在「搜尋」模式操作。根據預設，「傾聽」模式為預先設定。請調整您本地pand的行為。此外，請在YaST「網路卡」模組中設定bnepX介面(X代表系統中的設備號碼)。

手動設定藍芽

Bluez系統個別元件的組態檔案位於/etc/bluetooth目錄中。唯一的例外是啟動元件的檔案/etc/sysconfig/bluetooth，可由YaST模組修改。

以下所述的組態檔案僅可由使用者root修改。目前沒有變更「所有」設定值的圖形使用者介面。最重要的設定值可使用YaST「藍芽」模組來設定，如節17.2.2, “以YaST設定藍芽”中所述一般。所有其他的設定，僅在特殊狀況的資深使用者才會用到。但是，預設值通常就足夠。

PIN號碼提供基本保護，避免不需要的連線。行動電話在建立第一個連絡人時(或在設定電話的設備連絡人時)通常會查詢PIN。兩個設備若要通訊，兩者皆必須以相同PIN辨識本身。在電腦上，PIN位於檔案/etc/bluetooth/pin中。

藍芽連線的安全性

不管PIN，兩個設備之間的傳輸可能不是完全安全。根據預設，「藍芽」連線的驗證和加密是停用的。啟動驗證和加密可能會導致與某些「藍芽」設備進行通訊時發生問題。

可以在組態檔案/etc/bluetooth/hcid.conf中變更各種設定，如設備名稱和安全性模式。通常，預設值就足夠。檔案包含描述各種設定選項的註解。

在所包括的檔案中的兩個區段是指定為options和device。第一個區段包含hcid用以啟動的一般資訊。第二個區段則包含個別的本地「藍芽」設備的設定值。

options區段最重要的設定之一為security auto;。如果設為auto，hcid會嘗試使用本地PIN以進行內送連線。如果它失敗，它就會切換至none，仍然嘗試建立連線。若要加强安全性，此預設值應設為user，以確保使用者在每次建立連線時都會被要求輸入PIN。

在device區段中設定電腦顯示在另一端的名稱。設備類別(例如，Desktop、Laptop或Server)是在此區段中定義。驗證與加密也是在這裏啟用或關閉。

17.2.3 系統元件和公用程式

「藍芽」的操作性視各種服務間的互動而定。至少需要兩個背景精靈：`hcid` (*host controller interface*，主機控制器介面)，做為「藍芽」設備和控制它的介面；以及 `sdpd` (*service discovery protocol*，服務探索通訊協定)，設備可藉此找出主機提供何種服務。如果在啟動系統時沒有自動啓用它們，`hcid` 和 `sdpd` 均可使用 `rcbluetooth start` 指令來啓用。此指令必須以 `root` 的身份執行。

以下段落簡略描述可與「藍芽」搭配使用的最重要外圍程序工具。即使現在提供了各種控制「藍芽」的圖形元件，看一下這些程式還是非常值得的。

部份指令必須以 `root` 的身份執行。包括測試遠端設備連線的指令 `l2ping(device_address)`。

hcitool

`hcitool` 可用來判斷是否偵測到本地或遠端設備。`hcitool dev` 指令可列出本地設備。輸出會對每一個偵測到的本地設備以 `(interface_name) (device_address)` 格式產生一行。

遠端設備可使用 `hcitool inq` 指令來搜尋。每一個偵測到的設備會傳回三個值：設備位址、時鐘偏移值和設備等級。設備位址相當重要，因為其他指令會使用它來辨識目標設備。時鐘偏移值主要為技術層次的用途。等級以十六進位值指定設備類型和服務類型。

指令 `hcitoolname (device-address)` 可用來判斷遠端設備的設備名稱。如果是遠端電腦，類別和設備名稱會與它的 `/etc/bluetooth/hcid.conf` 檔案中的資訊一致。本地設備位址會產生錯誤輸出。

hciconfig

指令 `/usr/sbin/hciconfig` 傳遞本地設備的進一步資訊。如果 `hciconfig` 以無引數執行，輸出會顯示設備資訊，如設備名稱 (`hciX`)、實體設備位址 (以 `00:12:34:56:78` 格式表示的 12 位數字) 和關於傳輸資料量的資訊。

`hciconfig hci0 name` 顯示當您的電腦接收到遠端設備要求時所傳回的名稱。除了查詢本地設備的設定之外，`hciconfig` 尚可用來修改這些設定。例如，`hciconfig hci0 name TEST` 可將名稱設為 `TEST`。

sdptool

程式 `sdptool` 可用來檢查特定設備提供的服務。指

令 `sdptoolbrowse (device_address)` 會傳回設備的所有服務。`sdptool search (service_code)` 指令可用來搜尋特定服務。此指令會掃描所有可存取

的設備以尋找要求的服務。如果有一個設備提供服務，程式會印出設備傳回的完整服務名稱以及簡短描述。輸入sdptool 而無任何參數，則可檢視所有可能服務代碼清單。

17.2.4 圖形應用程式

在Konqueror 中，輸入URL bluetooth:/ 以列出本地和遠端「藍芽」設備。連接兩下設備可取得設備提供的服務之綜覽。如果您以滑鼠在特定服務上移動，瀏覽器的狀態列會顯示服務所使用的設定檔。如果您按一下服務，會開啓對話方塊，詢問要做什麼：儲存、使用服務(必須啓動應用程式才能執行)或取消動作。如果您不想要再顯示對話方塊，可是想要永遠執行選取的動作，可以標示核取方塊。對於某些服務，尚未提供支援。對於其他服務，必須安裝額外的套件。

17.2.5 範例

本小節提供兩個可能的「藍芽」案例典型範例。第一個範例教導如何透過「藍芽」建立兩個主機之間的網路連線。第二個範例則是教導電腦與手機之間的連線。

兩台主機之間的網路連線

在第一個範例中，將在兩台主機H1 和H2 之間建立網路連線。這兩台主機的「藍芽」設備位址為**baddr1** 和**baddr2** (如上述在兩台主機上以指令hcitooldev 所決定)。主機應該以IP 位址192.168.1.3 (H1) 和192.168.1.4 (H2) 來辨識。

藉由pand (Personal Area Networking, 個人區域網路) 的協助建立「藍芽」連線。下列指令必須以使用者root 的身份執行。描述重點在於「藍芽」特定動作，不提供網路指令(ip) 的詳細說明。

輸入指令pand -s 以啓動主機H1 的pand。然後，可使用指令pand-c (baddr1) 建立主機H2 的連線。如果您在其中一台主機上輸入ip link show 以列出可用的網路介面，輸出應該包含如下的項目：

```
bnep0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000  
link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

除了00:12:34:56:89:90，輸出應該包含本地設備位址**baddr1** 或**baddr2**。現在此介面必須指定IP 位址並啓用。在H1 上，可使用下列兩個指令來執行：

```
ip addr add 192.168.1.3/24 dev bnep0  
ip link set bnep0 up
```


在H2上：

```
ip addr add 192.168.1.4/24 dev bnep0
ip link set bnep0 up
```

現在H1可從H2的IP 192.168.1.3之下存取。使用指令ssh192.168.1.4從H1存取H2(假設H2執行sshd, 在SUSE LINUX中預設便會啓用)。指令ssh192.168.1.4也可以一般使用者執行。

從行動電話到電腦的資料傳輸

第二個範例說明如何將含有內建數位相機的行動電話所建立的相片傳輸到電腦(不產生多媒體訊息傳輸的額外成本)。雖然不同行動電話的功能表結構不同, 但程序通常相似。如有需要, 請參考您電話的手冊。此範例描述從Sony Ericsson行動電話傳輸相片到膝上型電腦。電腦上的Obex-Push服務必須為可用的, 且電腦必須授與行動電話存取權。在第一步驟中, 使膝上型電腦上的服務可用。可以藉由bluez-utils套件的opd精靈執行。以下列指令啓動精靈：

```
opd --mode OBEX --channel 10 --daemonize --path /tmp --sdp
```

使用了兩個重要的參數：--sdp向sdpd註冊服務, 而--path /tmp則是指定程式接收資料的儲存位置—在此例中是儲存於/tmp。您也可以指定其他您具有寫入存取權的目錄。

現在行動電話必須認識電腦。要這樣做, 開啓電話上的‘連線’功能表並選取‘藍芽’。如有需要, 在選取‘我的設備’之前按一下‘開啓’。選取‘新設備’並讓電話搜尋膝上型電腦。如果偵測到設備, 它的名稱會出現在顯示幕上。選取和膝上型電腦相關的設備。如果您遭遇到PIN查詢, 請輸入在/etc/bluetooth/pin中指定的PIN。現在您的電話可辨識膝上型電腦, 並且可和膝上型電腦交換資料。結束目前功能表並移至影像功能表。選取要傳輸的影像, 並按下‘其他’。在下一個功能表中, 按下‘傳送’以選取傳輸模式。選取‘透過藍芽’。膝上型電腦應該列為目標設備。選取膝上型電腦以啓動傳輸。然後影像會儲存在opd指令所指定的目錄。音軌可以相同方式傳輸到膝上型電腦。

17.2.6 疑難排解

如果您無法建立連線, 請根據下列清單繼續。請記住錯誤可能發生在連線的任一端或甚至在兩端。如果可能, 以另一個「藍芽」設備重新建構問題, 以確認設備沒有損壞。

本地設備是否列在`hcitooldev`的輸出中？

如果本地設備沒有列在此輸出中，表示`hcid`未啟動，或設備未被識別為「藍芽」設備。這可能有各種因素。設備可能損壞或正確的驅動程式可能遺失。內建「藍芽」的膝上型電腦通常有無線設備(如WLAN和「藍芽」)的開啓和關閉開關。請檢查您膝上型電腦的手冊以查看您的設備是否有此開關。以指令`rcbluetoothrestart`重新啟動「藍芽」系統，並檢查`/var/log/messages`中是否報告任何錯誤。

您的「藍芽」介面卡是否需要韌體檔案？

如果需要，請安裝`bluez-bluefw`並以指令`rcbluetoothrestart`重新啟動「藍芽」系統。

`hcitoolinq`的輸出是否傳回其他設備？

請測試此指令一次以上。連線可能受到干擾，因為其他設備也使用「藍芽」的頻帶。

PIN符合嗎？ 檢查電腦的PIN號碼(在`/etc/bluetooth/pin`中)是否符合目標設備的。

遠端設備可以「看到」您的電腦嗎？ 請嘗試從遠端設備建立連線。檢查此裝置是否看得到電腦。

可以建立網路連線嗎(請見範例1)？ 第一個範例(網路連線)可能因為數種原因而無法運作。例如，兩台電腦其中之一不支援ssh通訊協定。請嘗試`ping192.168.1.3`或`ping192.168.1.4`。如果可以，請檢查`sshd`是否在作用中。其他問題可能是兩個設備其中之一已經有網路設定，與範例中的位址`192.168.1.X`衝突。如果是此狀況，請嘗試不同的位址，如`10.123.1.2`和`10.123.1.3`。

膝上型電腦是否顯示為目標設備(範例2)？ 行動設備可識別膝上型電腦上的Obex-Push服務嗎？

在「我的設備」中，選取對應設備並檢視「服務」清單。如果沒有顯示Obex-Push(即使在更新清單之後)，則問題是因為膝上型電腦上的`opd`所造成。`opd`是否在作用中？您對特定目錄是否有寫入存取權？

第二個範例以相反方向是否能運作？ 如果安裝了`obexftp`套件，在部份設備上可使用`obexftp -b <device_address> -B 10 -p <image>`指令來執行。已測試數種Siemens和Sony Ericsson型號並發現其可以作用。請參考`/usr/share/doc/packages/obexftp`中套件的文件。

17.2.7 若需詳細資訊

「藍芽」的使用和組態的各種說明之廣泛綜覽可在<http://www.holtmann.org/linux/bluetooth/> 中找到。其他有用的資訊和說明：

- 「藍芽」通訊協定堆疊的正式howto 整合於核心中：<http://bluez.sourceforge.net/howto/index.html>
- 連線到PalmOS PDA：<http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html>

17.3 紅外線資料傳輸

IrDA (紅外線資料關聯) 是以紅外線進行無線通訊的產業標準。目前販售的許多膝上型電腦均配備有IrDA 相容接收器，可與其他設備(如印表機、數據機、LAN 或其他膝上型電腦) 通訊。傳輸速率範圍從2400 bps 到4 Mbps。

IrDA 操作模式有兩種。標準模式SIR 透過序列介面存取紅外線連接埠。此模式幾乎可在所有系統上使用，而且對於大部份需求已經夠用。較快的模式FIR 必須要有IrDA 晶片的特殊驅動程式。FIR 模式並不支援所有晶片類型，因為缺乏適當的驅動程式。請在電腦的BIOS 中設定想要的IrDA 模式。BIOS 也會顯示SIR 模式所使用的序列介面。

您可以在Werner Heuser 的IrDA how-to 中找到的IrDA 相關訊息，網址為<http://tuxmobil.org/Infrared-HOWTO/Infrared-HOWTO.html>。除此之外，您可以參考Linux IrDA Project 網站<http://irda.sourceforge.net/>。

17.3.1 軟體

所需的核心模組包括在核心套件中。irda 套件提供所需協助程式應用程式以支援紅外線介面。安裝套件之後，可在/usr/share/doc/packages/irda/README 找到文件。

17.3.2 組態

當系統啟動時，IrDA 系統服務不會自動啟動。請使用YaST IrDA 模組以啟動。在此模組中僅可修改一個設定：紅外線設備的序列介面。測試視窗會顯示兩個輸出。一個是irdadump 的輸出，記錄所有傳送和接收的IrDA 封包。此輸出應

包含電腦名稱和傳輸範圍中所有紅外線設備的名稱。這些訊息的範例顯示在節17.3.4, “疑難排解” 中。有IrDA 連線存在的所有設備列示於視窗下方。

IrDA 消耗相當可觀的電池電量，它每隔幾秒便傳送探索封包以偵測其他週邊設備。因此，如果您依賴電池電力，則應該僅在需要時啓動IrDA。輸入指令`rcirdastart` 以啓用或`rcirdastop` 以停用。在介面啓動時會自動載入需要的所有核心模組。

可在`/etc/sysconfig/irda` 檔案中執行手動組態。此檔案僅包含一個變數`IRDA_PORT`，它可以決定SIR 模式所使用的介面。

17.3.3 使用方式

資料可傳送到設備檔案`/dev/irlpt0` 以列印。設備檔案`/dev/irlpt0` 功能與一般`/dev/lp0` 有線介面類似，除了列印資料是以紅外線無線傳送之外。對於列印，請確認印表機在電腦紅外線介面的視覺範圍之內，以及紅外線支援已啓動。

透過紅外線介面操作的印表機可使用YaST 「印表機」 模組加以設定。因為不會自動偵測它，可以按一下「其他(未偵測到者)」 以手動設定它。在下列對話方塊中，選取「IrDA 印表機」。通常，`irlpt0` 是正確連線。在章12, 印表機操作中，可取得在Linux 中操作印表機的詳細資訊。

與其他主機和行動電話或其他類似設備的通訊，是透過設備檔案`/dev/ircomm0` 所指導。例如，Siemens S25 和Nokia 6210 行動電話可以使用紅外線介面，以`wvdial` 應用程式撥號並連線到網際網路。假使對應應用程式的設備設定已設定為`/dev/ircomm0`，也可以與Palm Pilot 同步化資料。

如果您想要，可以僅處理支援印表機或IrCOMM 通訊協定的設備。支援IROBEX 通訊協定的設備(如3Com Palm Pilot) 可使用特殊應用程式(如`irobexpalm` 和`irobexreceive`) 來存取。如需詳細資訊，請參考IR-HOWTO (<http://tldp.org/HOWTO/Infrared-HOWTO/>)。設備所支援的通訊協定是列在`irdadump` 輸出中設備名稱之後的括號內。IrLAN 通訊協定支援仍然是「進行中的工作」。

17.3.4 疑難排解

如果連線到紅外線連接埠的設備沒有回應，請使用指令`irdadump` (以`root` 的身份) 來檢查電腦是否辨識到其他設備。當Canon BJC-80 印表機在電腦的可見範圍中時，會固定出現類似範例17.1, “irdadump 的輸出” 的現象：

Example 17.1: irdadump 的輸出

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
                    hint=8804 [Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* earth
                    hint=0500 [ PnP Computer ] (21)
```

檢查介面的組態，是否沒有輸出或是其他設備沒有回覆。確認所使用的介面是正確的。紅外線介面有時候位在 `/dev/ttyS2` 或在 `/dev/ttyS3`，而有時候會使用IRQ 3 以外的中斷。這些設定幾乎可以在所有膝上型電腦的BIOS 設定功能表中檢查和修改。

簡單的視訊攝影機也可協助判斷紅外線LED 燈號是否全部都亮。大部份視訊攝影機可以看到紅外線；但是人眼卻不能。

熱插拔系統

熱插拔系統可以控制電腦中大部份設備的起始化。熱插拔系統不僅可用於作業期間插入和移除的設備，並可用於系統啓動時所偵測到的所有設備。它與sysfs檔案系統及udev 密切配合，在章19, 使用udev 的動態裝置節點中描述了相關的資訊。

18.1	設備和介面	294
18.2	熱插拔事件	295
18.3	熱插拔代辦	295
18.4	自動化模組載入	297
18.5	PCI 的熱插拔	297
18.6	開機程序檔冷插拔	298
18.7	錯誤分析	298

直到核心啓動時，只會起始必要的設備，例如匯流排系統、開機磁碟或鍵盤。核心會為所有偵測到的設備觸發熱插拔事件。udev精靈會傾聽這些事件，並呼叫個別的熱插拔程序檔以起始這些設備。對於無法自動偵測的設備或是在早期開機期間所遺失的事件，就會使用冷插拔。它會重播記錄的事件或掃描系統是否有未起始的設備，並為舊設備(像是ISA)使用靜態的組態設定。

除了少數的歷史特例，大部份的設備只要一提供存取，就會立即起始化，不論是在系統開機或熱插拔設備時。在起始化期間，會使用核心來註冊介面。此註冊將會接著觸發熱插拔事件，以對各自的介面進行自動設定。

在舊版的SUSE LINUX中，使用組態資料的靜態集做為起始設備的基礎。現在，系統會查看每個可用的設備，並搜尋適用的組態資料或產生它。

最重要的熱插拔功能設定在兩個檔案中。第一個檔案，`/etc/sysconfig/hotplug`，包含變數，可以影響hotplug和coldplug的行為。所有變數已加上註解。第二個檔案，`/proc/sys/kernel/hotplug`，包含由核心呼叫的執行程式名稱。設備組態位在`/etc/sysconfig/hardware`。從SUSE LINUX 9.3開始，這個檔案通常是空的，因為udev會經由網路連結插槽收到熱插拔訊息。

18.1 設備和介面

熱插拔系統不僅會管理設備，也會管理介面。設備會連結至匯流排或介面。匯流排可以視為一個多重介面。介面將設備連結至其他設備或應用程式。另外還有虛擬設備，例如，網路通道。設備的驅動程式通常必須是核心模組形式。介面大部份是由udev所建立的設備節點所代表。設備與介面的區別，對於瞭解整體概念很重要。

在sysfs檔案系統中所輸入的設備可以在`/sys/devices`之下找到。介面是位在`/sys/class`或`/sys/block`底下。sysfs中的所有介面應該有一個所屬設備的連結。不過，還是有部份設備不會自動新增此連結。沒有該連結，就無法知道此介面屬於哪一個設備，而且將無法找到合適的組態。

設備會透過設備描述來處理。這可以是sysfs的設備路徑(`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0`)、連接點的描述(`bus-pci-0000:02:00.0`)、個別ID(`id-32311AE03FB82538`)或其他類似的東西。在過去，介面是透過其名稱來定址。這些名稱代表現有設備的簡單編號，而且當新增或移除設備時有可能會變更。

介面也可以用相關設備的描述來定址。通常，內容會指出描述是指設備本身或它的介面。設備、介面和描述的典型範例包括：

PCI 網路卡 連接PCI匯流排的設備(`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0`或`bus-pci-0000:02:00.0`)，且具有網路介面(

(eth0, id-00:0d:60:7f:0b:22 或bus-pci-0000:02:00.0)。網路介面是由網路服務所使用，或者網路介面會連線至虛擬網路設備，例如通道或VLAN，因此會有介面。

PCI SCSI 控制器 設備(/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0 或bus-scsi-1:0:0:0)，以匯流排(/sys/class/scsi_host/host1)的形式，供多個實體設備介面使用。

SCSI 硬碟 設備(/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0 或bus-scsi-1:0:0:0)，包含多個介面(/sys/block/sda*)。

18.2 熱插拔事件

每一個設備和介面都有一個關聯的熱插拔事件，它是由udev及負責熱插拔代辦程式所處理。熱插拔事件是在建立或移除設備的連結，或是驅動程式註冊為介面或刪除介面時，由核心所觸發。從SUSE LINUX 9.3開始，udev會接收和分配熱插拔事件。udev會直接傾聽核心的網路連結訊息，或在/proc/sys/kernel/hotplug中必須指定/sbin/udevsend。在udev完成工作後(請參閱章19, 使用udev的動態裝置節點)，它會在/etc/hotplug.d/中搜尋符合事件類型的熱插拔代辦程式。

18.3 熱插拔代辦

熱插拔代辦是一個執行檔，執行事件合適的動作。設備事件的代辦程式位在/etc/hotplug.d/<事件名稱>與/etc/hotplug.d/default。將會依字母順序執行這些以.hotplug為字尾目錄中的所有程式。

若要確保忽略某一類型的事件，請從個別熱插拔代辦程式移除可執行位元。也可以將字尾.hotplug變更為其他的內容。

通常，設備代辦程式會載入核心模組，但偶而也會呼叫其他指令。在SUSE LINUX，這是由/sbin/hwup或/sbin/hwdown所處理。這些程式會在目錄/etc/sysconfig/hardware中，搜尋設備合適的組態，然後套用它。要防止特定設備起始，可建立合適的組態檔，並加上啟動模式manual或off。如果/sbin/hwup找不到任何組態，代辦會自動載入模組。在這個案例中的某些代辦程式會自動為hwup產生組態檔案。這將使代辦程式在下次執行時速度更快。如需更多資訊，請參閱節18.4, “自動化模組載入”。如需關於/sbin/hwup的詳細資訊，請參閱檔案/usr/share/doc/packages/sysconfig/README以及man hwup的手冊頁面。

在呼叫介面代辦程式前，udev 通常會先產生一個系統可以存取的設備節點。udev 可以指定持久性名稱給介面。如需詳細資料，請參閱章19, 使用udev 的動態裝置節點。接下來，個別代辦會設定介面。部份介面的程序敘述如下。

18.3.1 啟動網路介面

網路介面是使用/sbin/ifup 起始，而使用/sbin/ifdown 停止的。/usr/share/doc/packages/sysconfig/README 檔案以及man ifup 指令的手冊頁面提供詳細資訊。

如果電腦有多個不同驅動程式的網路設備，則在系統啟動時，如果另一個驅動程式載入速度比較快，則可能變更介面的指定。基於此因素，SUSE LINUX 會透過佇列的方式，管理PCI 網路設備的事件。在檔案/etc/sysconfig/hotplug 中使用變數HOTPLUG_PCI_QUEUE_NIC_EVENTS=no，就可以關閉此特性。

不過，最好的解決方案就是使用永久的介面指定。您可以在組態檔案中指定個別介面的名稱。如需此方法的詳細資訊，請參閱檔案/usr/share/doc/packages/sysconfig/README。從SUSE LINUX 9.3 開始，udev 也處理網路介面，雖然這些都不是設備節點。這將使永久介面名稱的使用更加標準化。

18.3.2 啟動儲存設備

必須裝載儲存設備的介面，才能存取它們。此可以完全自動化或預先設定。/etc/sysconfig/hotplug 和/etc/fstab 檔案中的變數HOTPLUG_DO_MOUNT、HOTPLUG_MOUNT_TYPE 和HOTPLUG_MOUNT_SYNC，會執行組態作業。完全自動化的作業可以藉由設定變數HOTPLUG_DO_MOUNT=yes 來啟動。若要停用它則可將變數設為no

使用變數HOTPLUG_MOUNT_TYPE，在兩個模式之間切換：subfs 或fstab。在HOTPLUG_MOUNT_TYPE=subfs 模式，子目錄是建立在/media 目錄中。子目錄的名稱是從設備特性衍生而來。存取媒體時，submountd 會自動裝載和解除裝載它。當設備不再可供存取時，就可以輕易移除此模式下的設備。在HOTPLUG_MOUNT_TYPE=fstab 模式下，儲存設備是使用傳統方法來裝載，即使用檔案/etc/fstab 中的合適項目。

變數HOTPLUG_MOUNT_SYNC 可以設定成在同步或非同步模式下開啓存取。在非同步模式下，寫入存取比較快，因為會緩衝結果。不過，不小心移除資料媒體會造成資料的不完整寫入。在同步模式下，所有資料都是立即寫入的，但是存取時間較長。設備必須使用umount 手動解除裝載。

建議使用持久性設備名稱，因為傳統設備名稱可能會因起始順序而變動。如需關於持久性設備名稱的詳細資訊，請參閱章19, 使用udev 的動態裝置節點章節。

18.4 自動化模組載入

如果設備無法使用 `/sbin/hwup` 起始，代辦程式會搜尋模組對應中合適的驅動程式。首先，它會搜尋包含在 `/etc/hotplug/*.handmap` 的對應。如果在該處找不到驅動程式，它也會搜尋 `/lib/modules/<kernelversion>/modules.*map`。若要使用核心的標準驅動程式之外的驅動程式，在 `/etc/hotplug/*.handmap` 輸入驅動程式，因為這是第一個讀取的檔案。

USB 代辦程式也會在 `/etc/hotplug/usb.usermap` 和 `/etc/hotplug/usb/*.usermap` 檔案中搜尋使用者模式驅動程式。使用者模式驅動程式是控制存取設備的程式，而不是控制存取核心模組的程式。使用此方法，可以呼叫特殊設備的執行檔。

如果是 PCI 設備，`pci.agent` 會先查詢驅動程式模組的相關 `hwinfo`。只有 `hwinfo` 不知道任何驅動程式的時候，代辦才會查尋 `pci.handmap` 以及核心對應。因為 `hwinfo` 已經在該處查看過了，所以查詢必定失敗。`hwinfo` 包含驅動程式對應的其他資料庫。不過，它也會載入 `pci.handmap` 來確定會套用此檔案的任何個別對應。

代辦 `pci.agent` 可能受限於來自 `/lib/modules/<kernelversion>/kernel/drivers` 中，特定子目錄的特定設備類型或驅動程式模組。第一個案例中，在檔案 `/usr/share/pci.ids` 最後找到的 PCI 設備，可以輸入檔案 `/etc/sysconfig/hotplug` 的變數 `HOTPLUG_PCI_CLASSES_WHITELIST` 和 `HOTPLUG_PCI_CLASSES_BLACKLIST`。至於第二個案例，在變數 `HOTPLUG_PCI_DRIVERTYPE_WHITELIST` 和 `HOTPLUG_PCI_DRIVERTYPE_BLACKLIST` 指定一或多個目錄。不會從排除的目錄中載入模組。在二個案例中，空的白名單會暗示所有的可能性，但排除黑名單的可能性。您也可以載入時排除單一模組。只要在 `/etc/hotplug/blacklist` 檔案中輸入模組，則代辦程式就不會載入該模組。在個別行寫入每一個模組名稱。

如果在對應檔案中找到多個合適的模組，只會載入第一個模組。要載入所有模組，請設定變數 `HOTPLUG_LOAD_MULTIPLE_MODULES=yes`。不過，最好為此設備建立個別的設備組態 `/etc/sysconfig/hardware/hwcfg-*`。

使用 `hwup` 載入的模組不受此影響。自動化模組載入只會在特例中發生，在未來的 SUSE LINUX 版本中會進一步的限制。但是如果找到合適的模組，代辦程式就會建立 `hwup` 組態檔案，以利下次使用。這可增加設備起始化的速度。

18.5 PCI 的熱插拔

部份電腦允許 PCI 設備的熱插拔。要充份利用此特點，必須載入特殊的核心模組。不過，這些模組會造成非 PCI 熱插拔電腦發生問題。很不幸地，無法自動偵測熱插拔 PCI 插槽。若要這樣做，請將 `/etc/sysconfig/hotplug` 檔案中的 `HOTPLUG_DO_REAL_PCI_HOTPLUG` 變數設定成 `yes`。

18.6 開機程序檔冷插拔

`boot.coldplug` 負責未自動偵測到的所有設備，以及未產生熱插拔事件的所有設備。它只是為指定為 `/etc/sysconfig/hardware/hwcfg-static-*` 的每一個靜態設備組態，呼叫 `hwup`。這也可以用來按照熱插拔的不同順序來起始內建設備，因為 `coldplug` 會在 `hotplug` 之前執行。

18.7 錯誤分析

18.7.1 日誌檔

除非另有指定，`hotplug` 只會傳送一些重要訊息至 `syslog`。要取得詳細資訊，將檔案 `/etc/sysconfig/hotplug` 中的變數 `HOTPLUG_DEBUG`，設定成 `yes`。如果將此變數設定成值 `max`，會為所有熱插拔程序檔，記錄每一個外圍程序指令。這表示在 `/var/log/messages` 中，用來儲存所有訊息的 `syslog` 會變得相當大。因為在開機程序中，直到啟動 `hotplug` 和 `coldplug` 之後，才會啟動 `syslog`，因此不會記錄第一個訊息。如果這些訊息對於您很重要，請透過變數 `HOTPLUG_SYSLOG` 指定其他日誌檔。如需關於此主題的資訊，請參閱 `/etc/sysconfig/hotplug`。

18.7.2 開機問題

如果在開機程序期間電腦當機，在開機提示輸入 `NOHOTPLUG=yes` 或 `NOCOLDPLUG=yes`，可以關閉 `hotplug` 或 `coldplug`。因為熱插拔停用的緣故，核心不會發出任何熱插拔事件。在執行中的系統，您可以輸入指令 `/etc/init.d/boot.hotplugstart`，啟動熱插拔。所有在該時間產生的事件，會發出並予以處理。若要拒絕排入佇列的事件，可以先在 `/proc/sys/kernel/hotplug` 中輸入 `bin/true`，以後再將項目重設成 `sbin/hotplug`。因為停用冷插拔之故，所以不會套用靜態組態。要套用靜態組態，您可以稍後輸入 `/etc/init.d/boot.coldplug start`。

要查明 `hotplug` 載入的特殊模組是否發生問題，請在啟動提示輸入 `HOTPLUG_TRACE=<N>`。所有要載入的模組名稱，在實際載入的 (N) 秒後，會列示在螢幕上。進行此作業時，您無法介入。

18.7.3 事件記錄器

`/sbin/hotplug` 會針對每個事件執行 `/sbin/hotplugeventrecorder` 程序檔。如果 `/events` 目錄存在，所有的熱插拔的事件都會在此目錄中儲存成個別

檔案。因此，可以針對測試用途來重新產生事件。如果此目錄不存在，不會記錄任何資料。

使用udev的動態裝置節點

Linux 核心2.6 為動態裝置目錄 `/dev` 引進新的使用者空間解決方案，並提供一致的裝置指定：`udev`。它僅提供實際出現的裝置檔案。它所建立或移除的裝置節點檔案通常是位於 `/dev` 目錄，而且它會重新命名網路介面。先前以 `devfs` 的 `/dev` 實作已不再使用，改由 `udev` 來取代。

19.1	建立規則	302
19.2	以NAME與SYMLINK自動化	303
19.3	關鍵字中的一般表示式	303
19.4	關鍵字選擇	303
19.5	大容量儲存裝置的一致名稱	304

傳統上，裝置節點會儲存在Linux系統上的/dev目錄中。每個裝置的可能類型都會有一個節點，無論其是否實際存在系統中。因此此目錄會佔用許多空間。udevfs指令帶來了一個重大的加強功能，因為現在只有當裝置真正存在，才會在/dev中給予一個裝置節點。

udev引進一個新的建立裝置節點的方法。它會以規則的形式，將sysfs所取得的資訊和使用者所提供資料進行比較。sysfs為核心2.6中的新檔案系統。它會提供關於連結系統的裝置的基本資訊。sysfs會裝載在/sys中。

使用者不需建立規則。如果連結了某個裝置，則會建立適當的裝置節點。不過，規則讓您得以變更節點的名稱。這讓您能便於用易記的名稱來取代難以理解的裝置名稱，也便於讓已連接相同類型的兩個裝置能有一致的裝置名稱。

除非有指定，否則會給予兩台印表機/dev/lp0與/dev/lp1的指定。哪個裝置會給予哪個裝置節點要視開啓印表機的順序而定。另一個範例是外接式大容量儲存裝置，例如USB硬碟。udev指令讓您可以在/etc/fstab中輸入確實的裝置路徑。

19.1 建立規則

在udev於/dev中建立裝置節點之前，會按照字母順序以字尾.rules讀取/etc/udev/rules.d中的所有檔案。會先使用符合裝置的第一個規則，即使其他規則也符合條件。井字符號(#)代表註解。規則採用以下格式：

```
key, [key,...]NAME [, SYMLINK]
```

至少必須指定一個關鍵字，因為規則被指定至裝置時，會以這些關鍵字為基礎。也必須指定一個名稱。建立在/dev中的裝置節點會使用此名稱。選擇性的符號連結參數讓節點可以建立在其他地方。印表機的規則可以採用以下格式：

```
BUS="usb", SYSFS{serial}="12345", NAME="lp_hp", SYMLINK="printers/hp"
```

在此範例中，有兩個關鍵字，BUS與SYSFS{serial}。udev會將此序號與連結到USB匯流排的裝置序號相比較。若要指定lp_hp名稱給/dev目錄中的裝置，所有的關鍵字都必須一致。此外，還會建立參照裝置節點的/dev/printers/hp符號連結。同時還會自動建立printers目錄。然後，列印工作便可以傳送到/dev/printers/hp或/dev/lp_hp中。

19.2 以NAME與SYMLINK自動化

NAME與SYMLINK參數能讓自動指派使用運算子。這些運算子會參照在對應裝置上的核心資料。用一個簡單的範例來說明該程序：

```
BUS="usb", SYSFS{vendor}="abc", SYSFS{model}="xyz", NAME="camera%n"
```

相機裝置的數字，例如camera0或camera1，會替代NAME中的%n運算子。另一個有用的運算子是%k，會被核心的標準裝置名稱所取代，例如hda1。您也可以使用udev規則呼叫外部程式，並使用以NAME和SYMLINK值傳回的字串。在man頁面中，可找到有關udev所有運算子的清單。

19.3 關鍵字中的一般表示式

在udev規則的關鍵字中，可以使用萬用字元來尋找符合的內容。例如，*字元就可以用來做為任何字元的保留字元，或?字元可以剛好做為一個字元的保留字元。

```
KERNEL="ts*", NAME="input/%k"
```

此規則會將標準目錄中的標準核心名稱指派給其指定的開頭字母為“ts”的裝置。如需關於在udev規則中使用一般表示式的詳細資訊，可以在man頁面manudev中找到。

19.4 關鍵字選擇

每個使用的udev規則必須要有良好的關鍵字。這裡有一些標準關鍵字的範例：

BUS 裝置匯流排類型

KERNEL 核心使用的裝置名稱

ID 匯流排上的裝置編號(例如，PCI匯流排ID)

PLACE 裝置所連接的實體點(如USB)

SYSFS{...} 像標籤、廠商、序號等的sysfs裝置屬性。

ID 與 PLACE 這兩個關鍵字可能很有用，但常使用的卻是 BUS、KERNEL 及 SYSFS{...}。udev 組態也會提供呼叫外部程序碼的關鍵字，並評估其結果。請在 `man udev` 中尋找關於此的詳細資訊。

sysfs 檔案系統會以目錄樹的方式來儲存有硬體資訊的小型檔案。每個檔案通常只包含一個項目的資訊，例如裝置名稱、廠商或序號。這些檔案每個都可以用來做為關鍵字值。不過，若要在一個規則中使用多個 SYSFS 關鍵字，您只能在相同目錄中使用檔案做為關鍵字值。udevinfo 工具可以協助找到有用的關鍵字值。

您必須找到一個參照相關裝置及包含 dev 檔案的 /sys 子目錄。這些目錄都位於 /sys/block 或 /sys/class 中。如果該裝置的裝置節點已經存在，udevinfo 可以為您找到正確的子目錄。udevinfo -q path -n /dev/sda 命令會輸出 /block/sda。這表示需要的目錄為 /sys/block/sda。現在用 udevinfo -a -p /sys/block/sda 指令來呼叫 udevinfo。也可以將兩個指令結合在一起，如 udevinfo -a -p `udevinfo -q path -n /dev/sda`。以下為輸出的摘要：

```
BUS="scsi"
ID="0:0:0:0"
SYSFS{detach_state}="0"
SYSFS{type}="0"
SYSFS{max_sectors}="240"
SYSFS{device_blocked}="0"
SYSFS{queue_depth}="1"
SYSFS{scsi_level}="3"
SYSFS{vendor}="          "
SYSFS{model}="USB 2.0M DSC      "
SYSFS{rev}="1.00"
SYSFS{online}="1"
```

請從輸出的資訊中，尋找將不會變更的適合關鍵字。請記住，通常您無法從不同的目錄使用關鍵字。

19.5 大容量儲存裝置的一致名稱

SUSE LINUX 內建有用的程序檔，能將相同的指定，指派至硬碟與其他儲存裝置，不論是要以何種順序來起始它們。/sbin/udev.get_persistent_device_name.sh 是一個包裝函式程序檔。首先它會呼叫 /sbin/udev.get_unique_hardware_path.sh，確定所指定裝置的硬體路徑。/sbin/udev.

`get_unique_drive_id.sh` 會擷取序號。接著，兩個輸出都會傳送到 `udev`，這能在 `/dev` 中建立裝置節點的符號連結。您可以在 `udev` 規則中直接使用包裝函式。以下為 SCSI 的範例，此範例也可歸納至 USB 或 IDE (將它寫成一行)：

```
BUS="scsi",  
PROGRAM="/sbin/udev.get_persistent_device_name.sh",  
NAME="%k", SYMLINK="%c{1+}"
```

當載入大容量儲存裝置的驅動程式後，它便會向核心註冊所有可用的硬碟。它們都會觸發呼叫 `udev` 的 `hotplug` 區塊事件。接著 `udev` 會讀取規則來判斷是否需要建立符號連結。

如果是透過 `initrd` 來載入驅動程式，則會遺失 `hotplug` 事件。不過，所有的資訊會儲存在 `sysfs` 中。`udevstart` 公用程式會在 `/sys/block` 與 `/sys/class` 中找到所有的裝置檔案，並啟動 `udev`。

還有一個 `boot.udev` 啟動程序檔，能在開機程序期間，重新建立所有的裝置節點。不過，啟動程序檔必須透過 YaST `runlevel` 編輯器來啟動，或用 `insserv boot.udev` 指令來啟動。

有一些工具與程式會認定 `/dev/sda` 是 SCSI 硬碟，`/dev/hda` 是 IDE 硬碟。如果與實況不符，這些程式便無法運作。YaST 會依賴這些工具，因此只能適用於核心裝置指定。

Linux 的檔案系統

Linux 支援多種不同的檔案系統。本章節簡要介紹最常用的Linux 檔案系統、詳盡闡述它們的設計概念、優點和應用範圍。同時並提供Linux 部份關於LFS (大型檔案支援) 的資訊。

20.1	詞彙	308
20.2	Linux 的主要檔案系統	308
20.3	其他支援的檔案系統	313
20.4	Linux 的大型檔案支援	314
20.5	若需詳細資訊	315

20.1 詞彙

中繼資料 一種檔案系統—確保磁碟上的所有資料都能適當組織而且可以存取的內部資料結構。基本上，它是“關於資料的資料。”幾乎每一種檔案系統都有自己的中繼資料結構，這也是檔案系統顯示不同效能特性的部份原因。它對於維護中繼資料的完整極為重要，因為要不是如此，檔案系統上所有資料便無法存取。

inode Inode 包含檔案的各種資訊，包括大小、連結數目、建立日期和時間、修改、存取，以及實際儲存檔案內容的磁碟區塊的指標。

日誌 在檔案系統的內容中，日誌是一種磁碟上的結構，包含檔案系統會在其中儲存對檔案系統的中繼資料進行何種變更的一種記錄。日誌處理大大降低Linux的復原時間，因為它放棄系統啓動時檢查整個檔案系統的冗長搜尋程序。相反地，只會QA日誌。

20.2 Linux 的主要檔案系統

不像兩、三年前，選擇Linux的檔案系統不再是幾秒鐘的事了(Ext2或ReiserFS?)。從2.4開始的核心，提供各種檔案系統可供選擇。下面是這些檔案系統基本工作方式以及其優點的概觀。

世上沒有符合所有應用程式類型的檔案系統，這一點很重要，要牢記於心。每一種檔案系統都有自己特殊的優、缺點，必須考慮在內。即使最尖端的檔案系統，也無法取代合理的備份策略。

資料整合性和資料一致性這兩個詞彙，用在本章節時，不是指使用者空間資料的一致性(應用程式寫入檔案中的資料)。此資料是否一致，必須由應用程式本身控制。

設定檔案系統

除非在本章節中提及，否則設定或變更分割以及檔案系統的一切步驟，都可以使用YaST模組來執行。

20.2.1 ReiserFS

ReiserFS 在 2.4 核心版本是正式的重要功能，而從 SUSE LINUX 6.4 版的 2.2.x SUSE 核心的核心修補程式就已經可以使用。ReiserFS 是由 Hans Reiser 和 Namesys 開發小組設計。ReiserFS 已經證實是舊有 Ext2 的強大替代方案。它的關鍵價值是較佳的磁碟空間利用、較好的磁碟存取效能以及快速損毀復原。不過，有一個小缺點：ReiserFS 太著重於中繼資料，對於資料本身則不重視。未來世代的 ReiserFS 會包括資料日誌處理(中繼資料和實際資料兩者都會寫入日誌)以及有序的寫入。

詳細說明 ReiserFS 的長處，包括：

更佳的磁碟空間利用 在 ReiserFS，所有資料按照稱為 B^{*}-平衡樹的結構整理的。樹狀結構提供更佳的磁碟空間利用，因為小的檔案可以直接儲存在 B^{*} 樹葉節點，而不是儲存在別處，而且只維護實際磁碟位置的指標。此外，未以 1 或 4 kB 的區塊配置儲存體，而是按需要的正確大小。另一項優點則依賴 inode 的動態配置。這樣會使得檔案系統比傳統的檔案系統更有彈性，例如在 Ext2，inode 密度必須在檔案系統建立期間指定。

更佳的磁碟存取效能 至於小的檔案，檔案資料和「stat_data」(inode) 資訊兩者通常儲存在一起。它們可以使用單一磁碟 I/O 作業來讀取，表示只需要存取一次磁碟，便能擷取所有需要的資訊。

快速損毀復原 使用日誌來追蹤最新的中繼資料變更，只要幾秒便能檢查檔案系統，即使很大的檔案系統也沒問題。

透過資料日誌的可靠性 ReiserFS 也支援資料日誌，以及 ordered 資料模式，此模式類似 Ext3 章節中(節 20.2.3, "Ext3") 描述的概念。預設模式為 data=ordered 可以確保資料和中繼資料的完整性，但是日誌僅適用中繼資料。

20.2.2 Ext2

Ext2 的起源要回到 Linux 歷史的古早年代。它的前輩- 延伸檔案系統，是在 1992 年 4 月落實並整合至 Linux 0.96c。延伸檔案系統已經過多次修改，而到了 Ext2，成為多年來最受歡迎的 Linux 檔案系統。有了日誌檔案系統的建立以及其驚人的快速回復時間，Ext2 就變得不那麼重要了。

簡短的 Ext2 功能摘要可以協助了解為什麼它(以及在某些領域) 是很多 Linux 使用者最喜愛的 Linux 檔案系統。

穩固性 Ext2 經過多次改良和密集測試，已經算是「老前輩」了。這可能是為什麼人們通常稱它堅如磐石的原因。在檔案系統無法完全取消裝載而導致系統中斷後，e2fsck 會開始分析檔案系統資料。中繼資料會進入一致性狀態而且待處理的檔案或資料區塊會寫入指定的目錄(稱為lost+found)。與日誌檔案系統相比，e2fsck 會分析整個檔案系統，不只是中繼資料最近修改的位元而已。這比檢查日誌檔案系統的記錄資料，要花費更多時間。按照檔案系統大小，此程序會花半小時或更長的時間。因此，不要為任何需要高可用性的伺服器選擇Ext2。不過，因為Ext2 不會維護日誌，而且使用相當少的記憶體，因此有時候比其他檔案系統較快速一些。

升級容易 Ext3 以Ext2 的程式碼做為強大的基礎，因此可以成為眾人喝采的下一代檔案系統。它的可靠性和穩固性，巧妙地結合了日誌檔案系統的優點。

20.2.3 Ext3

Ext3 是由Stephen Tweedie 設計。不像其他所有下一代檔案系統，Ext3 不依循全新的設計原則。它是以Ext2 為基礎。這兩個檔案系統彼此關係十分密切。Ext3 檔案系統可以輕易地在Ext2 檔案系統的最上層建立。Ext2 和Ext3 最重要的差別是Ext3 支援日誌處理。簡而言之，Ext3 提供三個主要優點：

可輕易從Ext2 升級，並具有很高的可靠性

因為Ext3 是以Ext2 程式碼為基礎，而且共用它的磁碟格式以及中繼資料格式，所以從Ext2 升級至Ext3 十分簡單。不像轉換至其他日誌檔案系統(例如ReiserFS、JFS 或XFS) 那麼冗長乏味(備份整個檔案系統，然後從頭建立)，轉換至Ext3 只是數分鐘的事。它也非常安全，因為重新建立整個檔案系統，不保證萬無一失。思考一下現有Ext2 系統等候升級至日誌檔案系統的數量，您就可以輕易地發現為什麼Ext3 對很多系統管理員具有一定的重要性。從Ext3 降級至Ext2 就和升級一樣容易。只要乾淨取消掛載Ext3 檔案系統，然後重新掛載成Ext2 檔案系統就可以了。

可靠性和效能 其他日誌檔案系統，有些會依照「僅中繼資料」日誌方法。這表示您的中繼資料永遠會保存在一致的狀態，但是同樣地無法自動保證檔案系統資料本身。Ext3 的設計是妥善管理中繼資料和資料二者。「管理」的程度可以自訂。在data=journal 模式啓用Ext3，可提供最大的安全性(資料整合性)，不過因為中繼資料和資料是記錄為日誌，所以系統速度會減慢。較新的方法是使用data=ordered 模式，可以確定資料和中繼資料整合性，不過限中繼資料使用日誌處理。檔案系統驅動程式會收集所有對應至某一中繼資料更新的所有資料區塊。更新中繼資料前，這些資料區塊會寫入硬碟。如此一來便可以達到中繼資料和資料的一致性，不會

犧牲效能。第三個要使用的選項是`data=writeback`，允許在其中繼資料已經提交至日誌後，將資料寫入主要檔案系統。一般認為此選項的效能最好。不過，它可以允許在損毀和復原舊資料後，重新顯示舊資料，同時又維護內部檔案系統整合性。除非您另有其他指定，否則Ext3 是預設與`data=ordered` 一起執行的。

20.2.4 將Ext2 檔案系統轉換成Ext3

從Ext2 轉換成Ext3，涉及兩個個別步驟：

建立日誌 以`root` 登入，然後執行`tune2fs -j`。這樣會以預設參數建立Ext3 日誌。要由自己決定日誌大小以及它所在的位置，執行`tune2fs -J` 取代想要的日誌選項`size=` 和`device=`。關於`tune2fs` 程式的詳細資訊，可以從它的手冊頁面(`tune2fs(8)`) 取得。

在/etc/fstab 指定檔案系統類型 要確定Ext3 檔案系統可以依此識別，請編輯檔案`/etc/fstab`，將指定給相應分割區的檔案系統類型從`ext2` 變更為`ext3`。所做的變更會在下次啟動時生效。

根目錄使用ext3 要啟動根目錄檔案設定為Ext3 分割區，請在`initrd` 中包括模組`ext3` 和`jbd`。要執行此步驟，請編輯檔案`/etc/sysconfig/kernel`，將這將這兩個模組包含在`INITRD_MODULES`，再執行指令`mkinitrd`。

20.2.5 Reiser4

核心2.6 發行後，有其他成員加入日誌檔案系統系列：Reiser4。Reiser4 與前一代產品ReiserFS (版本3.6) 全然不同。它導入調整檔案系統功能插件的概念，以及更詳細的安全性概念。

詳細的安全性概念 在設計Reiser4 方面，它的開發人員強調安全性相關功能的實作。因此Reiser4 會隨附一組專用的安全性插件。最重要的部分是導入檔案「項目」的概念。目前每個檔案都會定義檔案存取控制。如果大型檔案的資訊與多個使用者、群組或應用程式相關，則包含所有相關人員的存取權會相當不明確。在Reiser4 中，您可以將那些檔案分成較小部分的(「項目」)。可以分別設定每個項目和每位使用者的存取權，如此可擁有更明確的檔案安全性管理。完整的例子會是`/etc/passwd`。以前，只有`root` 可以讀取和編輯檔案，而非`root` 使用者僅有讀取該檔案的權限。使用Reiser4 的項目概念，您可以將該檔案分成一組項目(每位使用者一個項目)，讓使用者或應用程式修該他們自己的資料，但是不允許存取其他使用者的資料。此概念兼顧了安全性和彈性。

透過插件的擴充性 檔案系統會使用到的一般檔案系統功能和外部功能，Reiser 4 會以插件的方式執行。這些插件可以輕鬆地新增到基本系統。您不再需要重新編譯核心或重新格式化硬碟，就可以新增功能到您的檔案系統。

透過延緩的配置獲得較佳的檔案系統配置

Reiser4 和XFS 一樣，支援延緩的配置。請參閱節20.2.7, “XFS”。使用延緩的配置，就算是中繼資料，也可以產生較佳的整體配置。

20.2.6 JFS

JFS，是由IBM 開發的日誌檔案系統。JFS Linux 連接埠的第一個beta 版本，在2000 年夏天進入Linux 社群。1.0.0 版在2001 年問世。JFS 經過量身訂製後，可以符合以效能為最終目標的高運能伺服器的環境需求。JFS 是完整的64 位元檔案系統，可支援大型檔案以及分割區，這是它用在伺服器環境的另一個原因。

仔細觀察JFS 就知道為什麼這個檔案系統是您的Linux 伺服器不錯的選擇：

有效率的日誌處理 JFS 依照「僅中繼資料」方法。不需要密集檢查，只會檢查由最近檔案系統活動所產生的中繼資料變更，節省大量的復原時間。需要多個並行記錄項目的並行作業，可以結合成一個群組，大大降低透過多重寫入作業，所造成的檔案系統效能損失。

有效率的目錄組織 JFS 保存兩種不同的目錄組織。對於小型目錄，它允許目錄的內容直接儲存至它的inode。對於大型目錄，它使用B⁺樹，大大改善目錄管理。

透過動態inode 配置取得較佳的空間利用

以Ext2 為例，您必須事先定義inode 密度(管理資訊佔用的空間)，如此限制檔案系統的檔案或目錄數量上限。JFS 讓您不用擔心這些— 它動態配置inode 空間，在不需要的時候會釋放inode 空間。

20.2.7 XFS

1990 年代早期，SGI 開始對原先要當成IRIX OS 的檔案系統XFS 進行研發。XFS 背後的想法是建立高效能64 位元日誌檔案系統，配合今日嚴格的計算挑戰。XFS 對於操控大型檔案以及執行高階硬體，具備良好功能。不過，XFS 還是有一個缺點。和ReiserFS 一樣，XFS 在專注於中繼資料整合性，不重視資料整合性。

以下是XFS 關鍵特性的快速回顧，說明為什麼它可在高階計算中，對其他日誌檔案系統顯現強大的競爭能力。

利用配置群組，取得高擴充性。建立XFS 檔案系統時，檔案系統所屬的區塊裝置，會分割成8 或更多等同大小的線性區域。這些稱為配置群組。每一個配置群組管理自己的inode 以及可用的磁碟空間。事實上，配置群組可以看成是檔案系統中的檔案系統。因為配置群組彼此各自獨立，所以核心可以同時處理一個以上的配置群組。此特性是XFS 具有優良擴充性的關鍵。當然，獨立配置群組的概念符合多處理器系統的需求。

透過有效率的磁碟空間管理取得高效能

可用空間和inode 是由配置群組裡面的B⁺ 樹處理。使用B⁺ 樹可大大改良XFS 的效能和擴充性。XFS 會使用延緩的配置。它會將程序分成兩個部分來處理配置。待處理的交易會儲存在RAM 並保留適當的空間。XFS 仍然沒有決定資料到底要儲存在什麼地方(提到檔案系統區塊的時候)。此決策會盡量延緩至最後時刻。部分暫時資料永遠不會儲存至磁碟，因為當決定XFS 決定實際儲存位置時，它可能已經過時了。因此XFS 會增加寫入效能並降低檔案系統零散化。因為延緩的配置會造成寫入事件比其他檔案系統要來得少，因此寫入情況較嚴重時，發生損毀之後，有可能會遺失資料。

預先配置來避免檔案系統零散化。寫入資料至檔案系統前，XFS 會保留(預先配置) 檔案需要的可用空間。因此，可大幅降低檔案系統零散化。因為檔案的內容是分佈在檔案系統中，所以效能就會提高。

20.3 其他支援的檔案系統

表格20.1, “Linux 的檔案系統類型” 彙整Linux 支援的其他檔案系統。支援它們主要是確定不同媒體或外來作業系統中，資料交換的相容性。

Table 20.1: Linux 的檔案系統類型

cramfs	壓縮的ROM 檔案系統：ROM 的壓縮唯讀檔案系統。
hpfs	高效能檔案系統：IBM OS/2 標準檔案系統—僅支援唯讀模式。
iso9660	CD-ROM 的標準檔案系統。
minix	源自作業系統學術研究專案的檔案系統，是Linux 使用的第一個檔案系統。現在，它是磁片的檔案系統。
msdos	fat 最早源自DOS 的檔案系統，現在各種作業系統均使用之。

ncpfs	透過網路掛載Novell 卷冊的檔案系統。
nfs	網路檔案系統：使用這種檔案系統，資料可以儲存在網路的任何機器，而且可以經由授權從網路存取。
smbfs	一些產品(例如Windows) 使用伺服器訊息區塊透過網路存取檔案。
sysv	用於SCO UNIX、Xenix 和Coherent (個人電腦的商用UNIX 系統)。
ufs	用於BSD、SunOS 和NeXTstep。僅支援唯讀模式。
umsdos	MSDOS 的UNIX：套用在一一般fat 檔案系統之上，建立特殊檔案來達到UNIX 功能(許可權、連結、長檔案名稱)。
vfat	虛擬FAT：fat 檔案系統的副檔名(支援長檔名)。
ntfs	Windows NT 檔案系統，唯讀。

20.4 Linux 的大型檔案支援

一開始，Linux 支援的檔案大小最多是2 GB。在多媒體引爆之前，而且只要沒有人試著在Linux 操控大型資料庫，這已經夠用了。當應用程式必須使用的一組新介面時，修改核心和C 程式庫以支援超過2 GB 的檔案大小，對於伺服器計算變得越來越重要。現在，幾乎所有主要檔案系統提供LFS 支援，它讓您可以執行高階運算。表格20.2, “檔案系統的大小上限(磁碟上格式)” 提供您Linux 檔案和檔案系統目前限制的概觀。

Table 20.2: 檔案系統的大小上限(磁碟上格式)

檔案系統	檔案大小(位元組)	檔案系統大小(位元組)
Ext2 或Ext3 (1 kB 區塊大小)	2^{34} (16 GB)	2^{41} (2 TB)
Ext2 或Ext3 (2 kB 區塊大小)	2^{38} (256 GB)	2^{43} (8 TB)
Ext2 或Ext3 (4 kB 區塊大小)	2^{41} (2 TB)	2^{44} (16 TB)
Ext2 或Ext3 (8 kB 區塊大小)(含8 kB 頁面的系統，例如Alpha)	2^{46} (64 TB)	2^{45} (32 TB)

ReiserFS v3	2^{46} (64 GB)	2^{45} (32 TB)
XFS	2^{63} (8 EB)	2^{63} (8 EB)
JFS (512 位元組區塊大小)	2^{63} (8 EB)	2^{49} (512 TB)
JFS (4 kB 區塊大小)	2^{63} (8 EB)	2^{52} (4 PB)
NFSv2 (用戶端)	2^{31} (2 GB)	2^{63} (8 EB)
NFSv3 (用戶端)	2^{63} (8 EB)	2^{63} (8 EB)

Linux 核心限制

表格 20.2, “檔案系統的大小上限(磁碟上格式)” 敘述磁碟上格式的限制。2.6 核心採用自己的檔案大小限制以及所處理的檔案系統大小限制。限制如下：

檔案大小 在 32 位元系統，檔案不得超過 2 TB (2^{41} 位元組)。

檔案系統大小 檔案系統最大可以達 2^{73} 位元組。不過，此限制仍然跟不上目前可用的硬體。

20.5 若需詳細資訊

上述的每一個檔案系統專案，會維護自己的首頁，在其中可以找出郵件清單資訊、進一步文件以及常見問題。

- <http://e2fsprogs.sourceforge.net/>
- <http://www.zipworld.com.au/~akpm/linux/ext3/>
- <http://www.namesys.com/>
- <http://oss.software.ibm.com/developerworks/opensource/jfs/>

- <http://oss.sgi.com/projects/xfs/>

關於Linux系統深入的多部份教學課程，可以在IBM *developerWorks* 找到：<http://www-106.ibm.com/developerworks/library/l-fs.html>。關於Linux不同日誌檔案系統的比較，請參閱Juan I. Santos Florido在*Linuxgazette*的文章：<http://www.linuxgazette.com/issue55/florido.html>。對於深入分析Linux的LFS感興趣的人，應該參閱Andreas Jaeger的LFS網站：http://www.suse.de/~aj/linux_lfs.html。

使用PAM 驗證

Linux 在驗證程序中將PAM(可插式驗證模組，Pluggable Authentication Modules) 當做使用者與應用程式之間的溝通層。PAM 模組在整個系統都可使用，因此任何應用程式都可以要求它們。本章節說明模組驗證機制如何運作以及如何設定。

21.1	PAM 組態檔的結構	318
21.2	sshd 的PAM 組態	319
21.3	PAM 模組的組態	321
21.4	若需詳細資訊	323

系統管理員與程式設計人員通常會想要限制系統某些部份的存取，或是限制應用程式某些功能的使用。若未使用PAM，則每次引用新的驗證機制(例如LDAP或SAMBA)時，就必須調整應用程式。然而這個程序相當耗費時間，而且容易產生錯誤。避免這些缺點的方法就是將應用程式與驗證機制區隔開來，並將後者委託給集中管理的模組。如此一來每當需要新的驗證配置時，就能夠調整或撰寫適當的PAM 模組以供有問題的程式使用。

每個依賴PAM機制的程式都有自己的組態檔，位置在`/etc/pam.d/<programname>`目錄。這些檔案是定義用來驗證的PAM 模組。除此之外，在`/etc/security`之下具有大部份PAM 模組的全域組態檔，它們定義出這些模組的精確行為(這些範例包括`pam_env.conf`、`pam_pwcheck.conf`、`pam_unix2.conf` 以及`time.conf`)。每個應用程式使用PAM 模組時，其實就是呼叫一組PAM 函數，然後處理各種組態檔中的資訊，並將結果傳回呼叫的應用程式。

21.1 PAM 組態檔的結構

在PAM 組態檔中的每一行最多包含四個資料欄：

<模組類型> <控制旗標> <模組路徑> <選項>

PAM 模組是以堆疊的方式處理。不同模組的類型就有不同的目的，例如，一個模組檢查密碼，另一個驗證存取系統的位置，還有一個則讀取使用者特定的設定值。PAM 共有四種不同類型的模組：

auth 此類型模組的目的就是要檢查使用者的驗證性。傳統上是以查詢密碼的方式來檢查，但是這也可以使用晶片卡或透過生物測定(指紋或虹彩掃描)來完成。

account 此類型的模組會檢查使用者是否具有使用所要求服務的一般權限。舉例而言，執行這類檢查是要確保沒有人可以用使用者已過期的帳戶登入。

password 此類型模組的目的就是要能夠變更驗證權杖。在大部份情況下，這個權杖是密碼。

session 此類型的模組是用來管理和設定使用者的工作階段。模組在驗證之前和之後啟動，以便在系統日誌中註冊登入嘗試，並設定使用者的特定環境(郵件帳戶、主目錄、系統限制等等)。

第二個資料欄包含多種控制旗標，它們會影響已啓動模組的行為：

required 具有此旗標的模組必須成功地處理完成後，才能開始進行驗證。具有required 旗標的模組失敗後，需待所有具有相同旗標的其他模組處理完畢，使用者才會收到關於驗證嘗試失敗的訊息。

requisite 具有此旗標的模組也必須成功地處理，方式與具有required 旗標的模組相似。然而在失敗時，具有此旗標的模組會對使用者發出立即回應，而且不會再處理其他的模組。當成功時，就會接著處理其他的模組，像是任何具有required 旗標的模組。requisite 旗標可以當做基本的過濾器，檢查是否具備正確驗證所必需的條件。

sufficient 具有此旗標的模組經成功處理後，呼叫應用程式會收到關於成功的立即訊息，而且不會再處理其他的模組(假設具有required 旗標的模組之前沒有失敗)。具有sufficient 旗標的模組若失敗並不會有直接的結果，這是因為任何後續的模組都是以個別的順序處理。

optional 具有此旗標的模組不論是成功或失敗都不會有任何直接的結果。這對於只想顯示訊息(例如，通知使用者郵件已寄達)，但不想採取任何進一步動作的模組而言非常有用。

include 如果給予這個旗標，則指定為引數的檔案會在這個位置插入。

模組只要是位於/lib/security 的預設目錄中即可，其路徑並不需要明確指定，(至於SUSE LINUX 所支援的所有64 位元平台，目錄則為/lib64/security)。第四個資料欄可能包含指定模組的選項，例如debug (啓用除錯)或nullok (允許使用空密碼)。

21.2 sshd 的PAM 組態

為了顯示PAM 實際運作的理論，請將sshd 的PAM 組態當做實際範例：

Example 21.1: sshd 的PAM 組態

```
##PAM-1.0
auth    include      common-auth
auth    required     pam_nologin.so
account include     common-account
password include    common-password
session include     common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README.SuSE)
#session optional   pam_resmgr.so fake_ttyname
```

應用程式的一般PAM組態(在此例中為sshd)包含四個include陳述式，它們參照四個模組類型的組態檔：`common-auth`、`common-account`、`common-password`以及`common-session`。這四個檔案具有每個模組類型的預設組態。藉由包括它們而不是為每個PAM應用程式分別呼叫每個模組，這樣如果管理員變更預設值，就會自動取得更新的PAM組態。在以前，當PAM有變更或是安裝新應用程式時，必須為所有的應用程式手動調整所有的組態檔案。PAM組態以及所有對它所做的變更可以藉由預設組態檔來繼承。

第一個include檔(`common-auth`)會呼叫`auth`類型的兩個模組：`pam_env`與`pam_unix2`。請參閱範例21.2, “`auth` 區段的預設組態”。

Example 21.2: auth 區段的預設組態

```
auth    required    pam_env.so
auth    required    pam_unix2.so
```

第一個`pam_env`會載入`/etc/security/pam_env.conf`檔案，以根據此檔案的指定來設定環境變數。這可用來將`DISPLAY`變數設為正確的值，因為`pam_env`模組知道登入正在發生的位置。第一個模組`pam_unix2`會根據`/etc/passwd`與`/etc/shadow`檢查使用者的登入名稱與密碼。

在成功地呼叫`common-auth`中所指定的模組後，第三個模組會呼叫`pam_nologin`，以檢查`/etc/nologin`是否存在。如果它存在，則除了`root`以外的使用者都不能登入。在處理完`auth`模組的整個堆疊後，`sshd`才會收到關於登入是否成功的回應。假定堆疊的所有模組都有`required`控制旗標，則必須成功地處理完所有的模組後，`sshd`才會收到成功結果的訊息。如果其中一個模組沒有成功，仍然會處理整個模組堆疊，並且會在此時通知`sshd`失敗的結果。

只要已成功處理`auth`類型的所有模組，就會處理另一個include陳述式，在此例為範例21.3, “`account` 區段的預設組態”中的陳述式。`common-account`只包含一個模組：`pam_unix2`。如果`pam_unix2`傳回的結果是使用者存在，則`sshd`會收到此已成功的訊息，而且會接著處理下一個堆疊的模組(`password`)，如範例21.4, “`password` 區段的預設組態”中所示。

Example 21.3: account 區段的預設組態

```
account required    pam_unix2.so
```

Example 21.4: password 區段的預設組態

```
password required      pam_pwcheck.so  nullok
password required      pam_unix2.so    nullok use_first_pass use_authtok
#password required     pam_make.so     /var/yp
```

此外，sshd 的PAM 組態只有一個include 陳述式參照password 模組的預設組態，它是在common-password 中。不論應用程式何時要求變更驗證權杖，這些模組皆必須順利完成(控制旗標required)。變更密碼或另一個驗證權杖時，必需做安全性檢查。這是以pam_pwcheck 模組來達成。之後使用的pam_unix2 模組會沿用任何pam_pwcheck 的舊密碼或新密碼，因此使用者不必再進行驗證。這也使人無法規避pam_pwcheck 所執行的檢查。當account 或auth 類型前面的模組是設定成會針對過期密碼發出警告時，就應該使用password 類型的模組。

Example 21.5: session 區段的預設組態

```
session required      pam_limits.so
session required      pam_unix2.so
```

最後一個步驟會呼叫與common-session 檔案繫結在一起的session 類型的模組，以便根據有問題的使用者之設定值來設定工作階段。雖然會已再次處理pam_unix2，但是由於在此模組pam_unix2.conf 的個別組態中指定了none 選項，所以不會有實際的結果。pam_limits 模組會載入/etc/security/limits.conf 檔案，它可以定義某些系統資源的使用限制。當使用者登出時，會第二次呼叫session 模組。

21.3 PAM 模組的組態

有些PAM 模組是可設定的。對應的組態檔位於/etc/security。本小節簡單說明與sshd 範例相關的組態檔—pam_unix2.conf、pam_env.conf、pam_pwcheck.conf 以及limits.conf。

21.3.1 pam_unix2.conf

傳統的密碼驗證方法是由PAM 模組pam_unix2 所控制。它可以從/etc/passwd、/etc/shadow、NIS 對應、NIS+ 表格、或從LDAP 資料庫讀取所需的資料。設定個別應用程式本身的PAM 選項或編輯/etc/security/pam_unix2.conf 以進行全域設定，即可影響此模組的行為。範例21.6, "pam_unix2.conf" 中顯示了模組最基本組態檔。

Example 21.6: pam_unix2.conf

```
auth:    nullok
account:
password:    nullok
session:    none
```

auth 與 password 模組類型的 nullok 選項是指定對應的帳戶類型可以使用空密碼。也允許使用者變更帳戶密碼。session 模組類型的 none 選項是指定其本身不記錄訊息(這是預設值)。如需其他組態選項的資訊，請參閱檔案本身的註解以及 pam_unix2(8) 的手冊頁。

21.3.2 pam_env.conf

這個檔案可用來定義使用者的標準環境，每當呼叫 pam_env 模組時就會設定該環境。使用它即可使用下列語法預設環境變數：

```
VARIABLE [DEFAULT=[value]] [OVERRIDE=[value]]
```

VARIABLE 要設定的環境變數名稱。

[DEFAULT=[value]] 管理員想要設定的預設值。

[OVERRIDE=[value]] 可以由 pam_env 查詢及設定的值，用以覆寫預設值。

預設值應該由 pam_env 覆寫的最常見範例為 DISPLAY 變數，每當有遠端登入發生時它就會變更。請參閱範例 21.7, "pam_env.conf"。

Example 21.7: pam_env.conf

```
REMOTEHOST    DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
DISPLAY       DEFAULT=${REMOTEHOST}:0.0 OVERRIDE=${DISPLAY}
```

第一行將 REMOTEHOST 變數的值設為 localhost，每當 pam_env 無法決定任何其他值時，就會使用這個值。DISPLAY 變數接著就會包含 REMOTEHOST 的值。如需詳細資訊，可以從 /etc/security/pam_env.conf 檔案中的註解取得。

21.3.3 pam_pwcheck.conf

此組態檔是供pam_pwcheck 模組使用，該模組會從該檔案讀取所有password 類型模組的選項。此檔案所儲存設定值的優先順序，高於個別應用程式的PAM 設定值。如果尚未定義應用程式特定的設定值，應用程式就會使用全域設定值。範例21.8, “pam_pwcheck.conf” 說明pam_pwcheck 允許空的密碼和密碼的修改。模組的更多選項，請見/etc/security/pam_pwcheck.conf 檔案中的說明。

Example 21.8: pam_pwcheck.conf

```
password: nullok
```

21.3.4 limits.conf

系統限制可以以使用者或群組為單位，在limits.conf 檔案中做設定，該檔案是由pam_limits 模組讀取。該檔案允許您設定硬性限制，也就是完全不能超過，也可設定軟性限制，也就是可以暫時超過。若要瞭解關於語法以及可用選項的資訊，請參閱檔案中所包括的註解。

21.4 若需詳細資訊

在已安裝系統的/usr/share/doc/packages/pam 目錄中，可以找到下列的其他文件：

README 在此目錄的最上層中，有一些一般性的README 檔案。modules 子目錄中存有關於可用PAM 模組的README 檔案。

Linux-PAM 系統管理員指南 此文件包括系統管理員就PAM 所應該知道的每件事。它討論很多主題，涵蓋組態檔的語法至PAM 的安全性層面。該文件有三種格式：PDF 檔案、HTML 格式以及純文字。

Linux-PAM 模組撰寫者手冊 此文件是以開發人員的觀點將主題做成摘要，內含如何撰寫標準相容PAM 模組的資訊。它具有三種格式：PDF 檔案、HTML 格式以及純文字。

Linux-PAM 應用程式開發人員指南 想要使用PAM 程式庫的應用程式開發人員所需的全部內容都包括在此文件中。它具有三種格式：PDF 檔案、HTML 格式以及純文字。

Thorsten Kukuk 已開發一些適用於SUSE LINUX 的PAM 模組，並提供一些可用的相關資訊，網址為：<http://www.suse.de/~kukuk/pam/>。

Part III

服 務

基本網路

Linux 實際上是網際網路的產物，提供所有必要的網路工具及功能，可以整合到所有類型的網路結構。自訂的Linux 通訊協定、TCP/IP，具有各種服務與特殊功能，將會在這裏討論。使用網路卡、數據機或其他裝置進行網路存取可以使用YaST 來設定。也可使用手動方式來設定組態。本章節僅討論基本機制及相關的網路組態檔。

22.1	IP 位址與路由	330
22.2	IPv6 — 下一代的網際網路	333
22.3	名稱解析	339
22.4	使用YaST 手動設定網路連線	340
22.5	手動設定網路連線	349
22.6	smpppd 做為撥號助理	357

Linux 及其他 Unix 作業系統使用 TCP/IP 通訊協定。它不是單一網路通訊協定，而是能夠提供各種服務的網路通訊協定家族的一員。表格 22.1, “TCP/IP 通訊協定家族中的數種通訊協定” 中列示的通訊協定，提供透過 TCP/IP 在兩個機器之間交換資料的用途。由 TCP/IP、全球網路所結合而成的網路，整體上就是指「網際網路」。

RFC 代表要求建議(*Request for Comments*)。RFC 是描述作業系統及其應用程式的各種網際網路通訊協定和執行程序的文件。RFC 文件描述網際網路通訊協定的設定。若要擴展您對於任何通訊協定的知識，請參閱適當的 RFC 文件。這些文件可從 <http://www.ietf.org/rfc.html> 線上取得。

Table 22.1: TCP/IP 通訊協定家族中的數種通訊協定

協定	說明
TCP	傳輸控制通訊協定：以連線為導向的安全通訊協定。傳輸的資料首先由應用程式當做資料流傳送出去，然後再由作業系統轉換為適當格式。資料以最初傳送的原始資料流格式，抵達到目的地主機的相關應用程式。TCP 會決定傳輸期間是否遺失任何資料，且沒有混亂。只要是資料順序很重要的地方，就會執行 TCP。
UDP	使用者 Datagram 通訊協定：無連線、不安全的通訊協定。要傳送的資料以應用程式產生的封包形式加以傳送。不會保證資料在收件者端抵達的順序，而且可能遺失資料。UDP 適用以記錄為導向的應用程式。它的特點是延遲時間比 TCP 短。
ICMP	網際網路控制訊息通訊協定：基本上，這不是適用一般使用者的通訊協定，而是發佈錯誤報表的特殊控制通訊協定，能夠控制參與 TCP/IP 資料傳送之機器的行為。此外，它還提供特殊的回音模式，可以使用 ping 程式檢視。
IGMP	網際網路群組管理通訊協定：此通訊協定控制執行 IP 多重廣播時的機器行為。

如圖形 22.1, “TCP/IP 的簡化層模型” 中所顯示，資料交換發生在不同層。實際的「通訊層」是透過 IP (網際網路通訊協定, Internet Protocol) 進行不安全的資料傳輸。在 IP 的上方，TCP (傳輸控制通訊協定, Transmission Control Protocol) 可以保證資料傳輸某種程度的安全性。IP 層是由基本的硬體相依通訊協定所支援，例如乙太網路。

圖表提供每層的一或兩個範例。如您所見，層的順序是依據「抽象階

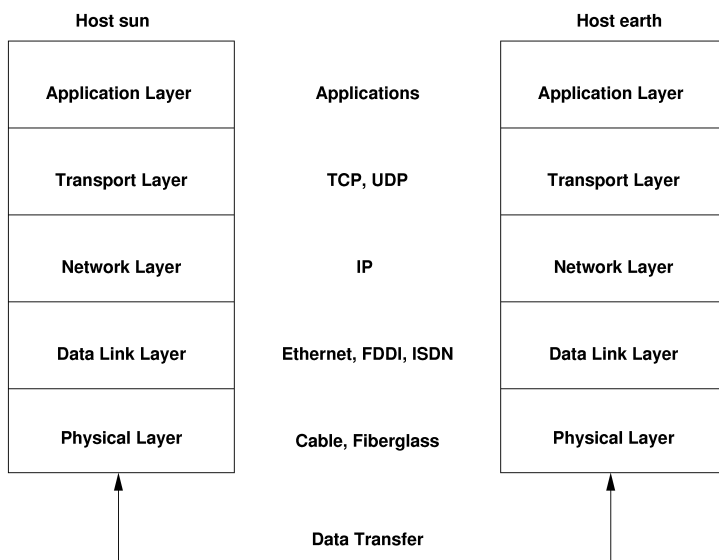


Figure 22.1: TCP/IP 的簡化層模型

層」(abstraction level)。最低層非常靠近硬體。不過，最上層對硬體而言幾乎是完全抽象的。每層都有自己的特殊功能。這些特殊功能通常隱含於其描述中。位元傳送及安全層代表使用的實體網路(如以太網路)。

幾乎所有的硬體通訊協定都是採用封包導向模式。要傳輸的資料是裝在封包中，因為無法一次傳送它。TCP/IP 封包的大小上限約為64 KB。封包一般而言較小，因為可能受限於網路硬體的關係。以太網路上資料封包的最大上限約為1500個位元組。在以太網路上傳送資料時，TCP/IP 封包的大小受限於此數量。如果傳送更多資料，則需要由作業系統傳送更多資料封包。

因為每層有自己指定的功能，關於每層的其他資訊必須儲存於資料封包中。這些資訊放在封包的「標題」中。每層皆在產生的封包前端附加小的資料區塊，稱為通訊協定標頭。在以太網路纜線上傳送的TCP/IP 資料封包範例，可參閱在圖形22.2, "TCP/IP 以太網路封包" 中的說明。proof sum 位於封包結尾，不在開頭處。這樣可幫助網路硬體簡化程序。

當應用程式在網路上傳送資料時，資料會經過每一層，除Layer 1 之外(網路卡)，全部執行於Linux 核心。每層都負責準備資料使其能夠傳送到下一層。最底層最後要負責傳送資料。接收到資料時則反轉執行整個程序。就如同洋蔥的層

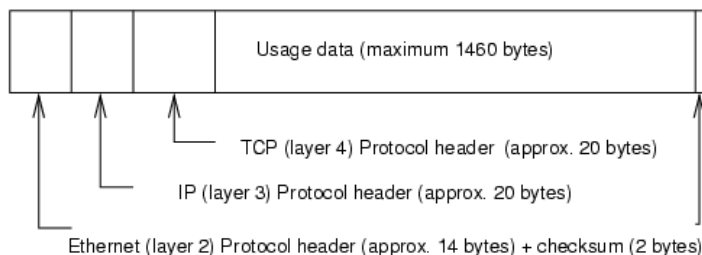


Figure 22.2: TCP/IP 乙太網路封包

次一般，在每層中，會從已傳輸的資料上移除通訊協定標頭。最後，Layer 4 負責讓目的地端的應用程式可以使用資料。以這種方式，每層僅直接與上下兩層通訊。對於應用程式而言，無論資料是透過 100 MBit/s FDDI 網路或 56-kbit/s 數據線進行傳輸，都沒有關係。同樣地，對於資料線而言，只要封包的格式正確，無論傳送的是哪種類型的資料也是無關的。

22.1 IP 位址與路由

在此節中的討論僅限於 IPv4 網路。如需有關 IPv6 通訊協定 (IPv4 的後繼者) 的資訊，請參閱節 22.2, “IPv6 — 下一代的網際網路”。

22.1.1 IP 位址

網際網路上的每台電腦都有唯一的 32 位元位址。這些 32 位元 (或 4 位元組) 一般所寫入的格式，如範例 22.1, “如何寫 IP 位址” 中的第二列所述。

Example 22.1: 如何寫 IP 位址

```
IP Address (binary):  11000000 10101000 00000000 00010100
IP Address (decimal):  192.    168.    0.    20
```

採用十進位格式，四位元組以十進位數字系統撰寫，以句號分隔。IP 位址是指定給主機或網路介面。其他地方無法使用。當然此規則一定有例外，但是在下文中只扮演微不足道的角色。

乙太網路卡本身有唯一位址，即MAC，或媒體存取控制位址(media access control address)。它的長度是48位元，在國際間是唯一的，而且由網路卡廠商設定在硬體上。不過，廠商指定位址的壞處是，—MAC位址不會組成階層系統，而是多少有些隨機散佈。因此，無法用來定位遠端機器的位址。MAC位址在區域網路中主機之間的通訊，仍然扮演了重要的角色，而且是Layer 2通訊協定標題的主要元件。

IP位址中的點表示階層系統。直到1990年代，IP位址仍嚴格地以類別加以分類。然而，此系統已證明太過死板而停止採用了。現在，則是使用「無類別路由」(classless routing)，即CIDR(無類別領域間路由，classless interdomain routing)。

22.1.2 網路遮罩與路由

網路遮罩的構想，目的是通知具有IP位址192.168.0.1的主機，其具有IP位址192.168.0.20主機的位置。簡單來說，具有IP位址主機上的網路遮罩，定義什麼是內部和外部。位於內部(「在相同子網路」)的主機會直接回應。位於外部(「不在相同子網路」)內的主機僅透過閘道或路由器回應。因為每個網路介面可以接收自己的IP位址，因此可能變得相當複雜。

傳送網路封包前，會在電腦上執行以下程序。IP位址透過邏輯AND連結到網路遮罩，而傳送主機的位置也是透過邏輯AND連接到網路遮罩。如果有數個可用的網路介面，通常會驗證所有可能的傳送者位址。會比較AND連結的結果。如果比較後是相符的，表示目的地或接收主機位於相同的子網路中。否則，必須透過閘道加以存取。愈多的“1”位元位於網路遮罩中，則可以直接存取的主機愈少，而透過閘道才能達到的主機愈多。在範例22.2, “連結IP位址到網路遮罩”中說明了幾個範例。

Example 22.2: 連結IP位址到網路遮罩

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0): 11111111 11111111 11111111 00000000
-----
Result of the link:      11000000 10101000 00000000 00000000
In the decimal system:   192.    168.    0.      0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask   (255.255.255.0): 11111111 11111111 11111111 00000000
-----
Result of the link:      11010101 10111111 00001111 00000000
In the decimal system:   213.    95.    15.    0
```

顯示的網路遮罩如IP位址一樣，採用十進位制，以句號分隔。由於網路遮罩也是32位元值，因此是以彼此相鄰的四個數值來表示。必須設定可以在哪些網路介面上存取哪些閘道與位址領域。

另一個範例：使用相同乙太網路纜線連接的所有機器，通常位於同一個子網路中，而且可以直接存取。以交換器或橋接器分配乙太網路時，仍然可以連接這些主機。

可是，經濟型的乙太網路不適合涵蓋較大的距離。您必須傳輸IP封包到其他硬體，例如FDDI或ISDN。此傳輸所用的設備稱為路由器或閘道。Linux機器可以完成這項工作。相關的選項稱為ip_forwarding。

如果已經設定閘道，會傳送IP封包到適當的閘道。在到達目的地主機或封包的TTL(持續時間)過期前，會嘗試以相同的方式(從主機到主機)轉送封包。

Table 22.2: 特定位址

位址類型	說明
基本的網路位址	這是網路遮罩「及」網路中的任何位址，如Result下的範例22.2, “連結IP位址到網路遮罩”所顯示。此位址不能指定給任何主機。
廣播位址	基本來說，即為「存取此子網路的所有主機」。若要產生此位址，網路遮罩會以二進位格式反轉，連結到具有邏輯OR的基本網路位址。因此上述範例會成為192.168.0.255。此位址不能指定給任何主機。
本地主機	位址127.0.0.1是嚴格指定到每個主機上的「迴路設備」(loopback device)。使用此位址可以設定到您自己機器的連線。

因為IP位址在全世界必須是唯一的，您不能只選取隨機位址。有三種位址領域可以用來設立私人IP架構的網路。利用這些位址，無法設定到其他的網際網路連線，除非您套用特定技巧，因為這些位址無法在網際網路上傳輸。這些位址領域指定於RFC 1597並列於表格22.3, “私人IP位址領域”中。

Table 22.3: 私人IP位址領域

網路/網路遮罩	領域
10.0.0.0/255.0.0.0	10.x.x.x
172.16.0.0/255.240.0.0	172.16.x.x-172.31.x.x

192.168.0.0/255.255.0.0 192.168.x.x

22.2 IPv6 — 下一代的網際網路

由於全球資訊網(World Wide Web, WWW)的出現，過去十五年內，在網際網路上透過TCP/IP進行通訊的電腦數目暴增。自從CERN的Tim Berners-Lee (<http://public.web.cern.ch>)於1990年發明WWW以來，網際網路主機的數量從幾千台成長為幾百萬台。

如前面所述，IP位址僅由32個位元組成。而且，損失了一些IP位址—由於組織網路的方式，使得這些IP位址無法使用。您的子網路中可用的位址數目是位元數的平方減2。例如，子網路有2個、6個或14個位址可用。例如，如果要連接128個主機到網際網路，則子網路需要256個IP位址，但是其中只有254個可用，因為子網路結構本身需要用掉兩個IP位址：即廣播位址與基本網路位址。

在目前的IPv4通訊協定之下，DHCP或NAT(網路位址轉譯，Network Address Translation)是典型的機制，可用來避免位址可能不足的問題。搭配保持私人和公用位址空間分開的方式，能夠減輕短少的情形。其中產生的問題是在於其組態，設定麻煩且難於維護。若要在IPv4網路中設定主機，需要一些位址項目，如主機自己的IP位址、子網路遮罩、閘道位址，可能還需要名稱伺服器位址。您必須知道所有這些項目，且無法從其他地方取得。

透過IPv6，位址短少及繁複組態的情形應該都成為過去式了。以下小節說明更多IPv6改善的部分及它帶來的好處，還有關於從舊通訊協定轉移到新通訊協定的資訊。

22.2.1 優點

新通訊協定帶來的最重要、最顯而易見的改善，是能夠大量擴充可用的位址空間。IPv6位址是由128個位元值組成，而不是傳統的32位元。這樣提供了數以千兆的IP位址。

然而，IPv6位址不僅是在長度方面與之前的位址不同；這些位置的內部結構也不同，可能包含有關系統及其所屬網路的更明確資訊。有關IPv6的詳細資訊，可以在節22.2.2, “定址類型與結構”中找到。

以下是新通訊協定一些其他優勢的清單：

自動設定 IPv6讓網路能夠「隨插即用」(plug and play)，表示新設定的系統不需經過任何手動設定，即可整合到(區域)網路。新主機使用其自動設定組

態機制，從鄰近的路由器上可用的資訊取得自己的位址，依賴的是稱爲「網路芳鄰尋找」(*Neighbor Discovery, ND*)的通訊協定。這個方法不需要管理員的介入，而且不需要維護分配位址的中央伺服器，這是IPv4的另一個優勢，因爲自動位址分配需要DHCP伺服器。

機動性 IPv6能夠同時將數個位址指定給一個網路介面。這樣使用者就能輕鬆地存取數個網路，可以比擬爲行動電話公司提供的國際漫遊服務：當您帶著手機出國時，電話只要進入通信區域，就會自動登入國外的服務，如此一來，使用同樣的電話號碼仍然可以找到您，而且您也可以撥打電話，就像在國內一樣。

安全通訊 使用IPv4，網路安全性是附加的功能。IPv6包括IPSec爲其中一個核心功能，允許系統在安全的通道上進行通訊，避免網際網路上的外人竊聽。

舊版相容性 實際上，不可能一次將整個網際網路從IPv4切換到IPv6。因此，很重要的是，兩個通訊協定不僅要能夠共存於網際網路上，也得要能夠共存於一個系統中。這是藉由相容的位址(IPv4位址可以輕易轉譯爲IPv6位址)以及使用一些通道來確保共存。請參閱節22.2.3, “IPv4與IPv6的共存”。另外，系統可以仰賴「雙重堆疊IP」(*Dual Stack IP*)技術，同時支援這兩種通訊協定，這表示系統有兩個完全分開的網路堆疊，如此一來，兩種通訊協定版本不會相互干擾。

透過多重廣播自訂量身訂做的服務 利用IPv4，有些服務(如SMB)需要廣播它們的封包到區域網路上的所有主機。IPv6以更精細的方法，透過「多重廣播」(*multicasting*)—將一些主機定位爲群組的一部分，讓伺服器定址主機(這與透過「廣播」(*broadcasting*)定位所有主機，或透過「單點廣播」(*unicasting*)個別定址每個主機的方式不同)。定址爲群組的主機，取決於具體的應用程式。例如，有些預先定義的群組可以定址所有名稱伺服器(「所有名稱伺服器多重廣播群組」)或所有路由器(「所有路由器多重廣播群組」)。

22.2.2 定址類型與結構

如前面所提，目前的IP通訊協定欠缺兩個重要層面：IP位址正在逐漸短缺，而設定網路及維護路由表的工作變得更加複雜而繁重。IPv6透過擴充位址空間到128個位元解決了第一個問題。第二個問題的解決方式則是引入階層位址結構，結合配置網路位址的複雜技術，以及*multihoming* (指定數個位址到一個設備，可以存取數個網路)。

使用IPv6時，瞭解三種不同類型的位址是很有用的：

單點廣播(Unicast) 這類位址恰好與一個網路介面有關聯。這類位址的封包僅傳送到一個目的地。因此，單點廣播位址用來傳送封包到區域網路或網際網路上的個別主機。

多重廣播(Multicast) 這類位址與一組網路介面有關聯。這類位址的封包會傳送到屬於該組的所有目的地。多重廣播位址主要由特定網路服務使用，可直接與特定主機群組通訊。

任點廣播(Anycast) 這類位址與一組介面有關聯。這類位址的封包會根據基礎路由通訊協定的原則，傳送到最靠近傳送者的群組成員。使用任點廣播位址，讓主機更易於找出在指定網路區域中提供特定服務的伺服器。相同類型的所有伺服器擁有一樣的任點廣播位址。只要主機要求服務，它會從最靠近位置的伺服器接收回覆，由路由通訊協定決定。如果此伺服器因為某種原因失敗，通訊協定會自動選取第二個最靠近的伺服器，或是選取第三個伺服器，依此類推。

IPv6 位址由八個四位數欄位組成，每個都代表16個位元，以十六進位標記法寫入。這些位址以冒號(:)分隔。指定欄位內的任何前導零位元組可以刪除，但是欄位內或尾端的零不能刪除。另一個慣例是多於四個連續的零位元組可能會摺疊成兩個冒號。不過，每個位址僅允許一個這類的::。這類的簡略的標記法，顯示於範例22.3, “範例IPv6 位址”中，其中三行都是代表相同的位址。

Example 22.3: 範例IPv6 位址

```
fe80 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                               : 10 : 1000 : 1a4
```

IPv6 位址的每個部分都有定義的功能。第一個位元組形成字首，指定位址類型。中間的部分是位址的網路部分，但是可能不會使用。位址的尾端形成主機部分。透過IPv6，在位址尾端的斜線後表示字首的長度，可以定義網路遮罩。位址如範例22.4, “指定字首長度的IPv6 位址”中所示，包含的資訊是形成位址網路部分的前64個位元以及形成其主機部分的最後64個位元。換句話說，64表示網路遮罩從左邊開始填入64個1位元值。就像IPv4一樣，IP位址使用AND結合網路遮罩的值，判斷主機是否位於相同的子網路或另一個子網路。

Example 22.4: 指定字首長度的IPv6 位址

```
fe80::10:1000:1a4/64
```

IPv6 知道關於數個字首的預定類型。有一些顯示在表格22.4, “各種IPv6 字首”中。

Table 22.4: 各種IPv6 字首

字首(十六進位)	定義
00	IPv4 位址與透過IPv6 的IPv4 相容位址。這些位址用來維護與IPv4 的相容性。其使用仍然要求路由器能夠轉譯IPv6 封包為IPv4 封包。數個特殊的位址，如迴路設備的位址，也有此字首。
2 或3 做爲第一個數字	可彙總的全域單點廣播位址。在IPv4 的情形中，可以指定介面形成部分的特定子網路。目前而言，有以下位址空間：2001::/16 (製造品質位址空間) 與2002::/16 (6to4 位址空間)。
fe80::/10	連結本地位址。具有這種字首的位址不應該傳送，因此僅能從相同的子網路內連接。
fec0::/10	本地網站位址。這些位址可以傳送，但是僅能在所屬組織的網路內傳送。事實上，它們等同於目前私人網路位址空間的IPv6 (例如，10.x.x.x)。
ff	這些都是多重廣播位址。

單點廣播位址由三個基本元件組成：

公用拓撲(Public Topology) 第一個部分(也包含上述的其中一種字首) 用來透過公用網際網路傳送封包。它包含了有關提供網際網路存取的公司或機構資訊。

網站拓撲(Site Topology) 第二個部分包含有關傳送封包的目的地子網路的傳送資訊。

介面識別碼(Interface ID) 第三個部分識別傳送封包的介面。它也允許MAC 形成部分的位址。這個前提是MAC 在全球是唯一的，由硬體製造商在設備中編碼固定識別碼，可相當程度地簡化組態程序。事實上，結合前64 個位址位元形成EUI-64 記號，加上從MAC 取得的最後48 個位元，而其餘的24 個位元則包含有關記號類型的特殊資訊。如此一來，就可以指定EUI-64 記號給沒有MAC 的介面，例如以PPP 或ISDN 為基礎的介面。

在此基本結構的最上層，IPv6 會分辨五種不同類型的單點廣播位址：

:: (未指定的) 第一次啓始介面時，如果還未用其他方法判定位址時，主機會使用此位址做爲其來源位址。

::1 (迴路) 迴路設備的位址。

IPv4 相容位址 IPv6 位址是由IPv4 位址以及由96 個零位元組成的字首形成的。這類相容性位址用於通道(請參閱節22.2.3, “IPv4 與IPv6 的共存”), 允許IPv4 與IPv6 主機在純IPv4 環境中彼此通訊。

對應到IPv6 的IPv4 位址 這類位址以IPv6 標記法指定純IPv4 位址。

本地位址 有兩種位址類型用於本地：

連結本地 這類位址僅能用於本地子網路。具有此類型來源或目標位址的封包不應該傳送到網際網路或其他子網路。這些位址包含特殊字首(fe80::/10) 以及網路卡的介面識別碼, 加上由空位元組所組成的中間部分。自動設定組態以便與屬於相同子網路中的其他主機通訊時, 會使用這類位址。

網站本地 具有這種位址的封包可以傳送到其他子網路, 但是不能到更寬廣的網際網路, 而必須保留在組織自己的網路內。這類位址用於內部網路, 而且等同於IPv4 所定義的私人位址空間。它們包含特殊字首(fec0::/10)、介面識別碼, 以及指定子網路識別碼的16 位元欄位。同樣地, 其他則是填入空位元組。

因為引入了IPv6 這種全新的功能, 所以每個網路介面通常會取得數個IP 位址, 其優點是可透過相同介面存取數個網路。其中一個網路可以設定完全自動化(使用MAC 和已知的字首), 只要一啓用IPv6 (使用連結本地位址) 即可連接區域網路上的所有主機。利用形成位址部分的MAC, 全球使用的任何IP 位址都成為唯一的。位址的唯一變數部分, 是指定「網站拓撲」和「公用拓撲」, 該部分視主機目前正在操作的實際網路而定。

如果主機要在不同的網路之間往返, 至少需要兩個位址。其中一個, 即「主位址」, 不僅包含了介面識別碼, 也包含了其通常所屬之主網路(及其對應字首) 的識別碼。主位址是靜態位址, 因此它通常不會變更。儘管如此, 預定要送到行動主機的所有封包, 還是可以傳送到主位址, 無論是在主網路或其他外部網路中操作。這可藉由IPv6 全新功能來達成, 如「無狀態自動設定」與「網路芳鄰尋找」。除了其主位址外, 行動主機也取得一或多個其他的位址, 這些位址屬於漫遊的外部網路。這些外部網路稱為*care-of* 位址。主網路具有封包在外部漫遊時轉寄預定要送到主機的設備。在IPv6 環境中, 這個工作是由「主代辦」執行的, 它會取得所有預定要送到主位址的封包, 透過通道轉送它們。另一方面, 預定送到*care-of* 位址的封包會直接傳送到行動主機, 不會特別繞行。

22.2.3 IPv4 與IPv6 的共存

連接網際網路的所有主機從IPv4 轉移到IPv6 是一種漸進程序。這兩種通訊協定某些時候會共存。在一個系統上共存，可保證執行兩種通訊協定的「**雙重堆疊**」。但是仍出現一些問題，就是使用IPv6 的主機如何與IPv4 主機通訊，以及由IPv4 架構主導的目前網路如何傳輸IPv6 封包。最佳的解決方案是提供通道及相容性位址(請參閱節22.2.2, “定址類型與結構”)。

在(全球) IPv4 網路中多少有些隔離的IPv6 主機可以透過通道進行通訊：IPv6 封包會封裝成IPv4 封包，在IPv4 網路上移動。兩個IPv4 主機之間的連線，稱為「**通道**」。若要完成這個目的，封包必須包含IPv6 目的地位址(或對應字首)以及通道接收端上遠端主機的IPv4 位址。基本通道可以根據主機管理員之間的協議「**手動**」設定；這也稱為「**靜態通道**」。

不過，靜態通道的組態及維護通常需要密集勞力，才能使用它們應付每天的通訊需求。因此，IPv6 提供三種不同的「**動態通道**」方法：

6over4 IPv6 封包會自動封裝成IPv4 封包，透過能夠多重廣播的IPv4 網路進行傳送。IPv6 的訣竅是將整個網路(網際網路)視為一個大型的區域網路(LAN)。如此即能自動判定IPv4 通道的接收端。然而，這個方法不能適當的延伸，而且也因IP 多重廣播目前在網際網路上並不普遍的事實而受到阻礙。所以，它僅能為啓用多重廣播的小型公司或機構的網路提供解決方案。這個方法的規格詳述於RFC 2529。

6to4 利用此方法，IPv4 位址會自動從IPv6 位址產生，使得隔離的IPv6 主機能夠在IPv4 網路上通訊。不過，有關這些隔離的IPv6 主機及網際網路之間的通訊，目前已出現一些問題。該方法詳述於RFC 3056。

IPv6 通道代理 這個方法仰賴提供IPv6 主機專屬通道的特殊伺服器。詳述於RFC 3053。

6bone 方案

「過去」網際網路的核心，已經有透過通道連接的IPv6 子網路的全域散佈網路。這就是6bone 網路(<http://www.6bone.net>)，這是IPv6 測試環境，由想要開發及提供IPv6 架構服務的程式人員及網際網路提供者使用，可獲得執行新通訊協定所需的經驗。詳細資訊請參閱專家的網際網路網站。

22.2.4 設定IPv6

若要設定IPv6，您通常不需要在個別工作站中做任何變更。不過，必須載入IPv6支援。如果要這樣做，請輸入`modprobe ipv6` 做為root。

由於IPv6的自動組態概念，網路卡會在連結本地網路中指定一個位址。工作站通常不會進行路由表格管理。工作站可使用「**路由器通告通訊協定**」，向網路路由器查詢應使用的前置號碼和閘道。可使用`radvd`程式來設定IPv6路由器。此程式會通知工作站該IPv6位址應使用的前置號碼和路由器。或者，也可使用`zebra`自動設定位址和路由的組態。

請參閱`ifup`的手冊頁(`man ifup`)，以瞭解如何使用`/etc/sysconfig/network`檔案來設定不同類型的通道。

22.2.5 若需詳細資訊

上述概觀沒有完整地涵蓋IPv6主題。如需更深入的探討這種新的通訊協定，請參閱以下線上文件和書籍：

<http://www.ngnet.it/e/cosa-ipv6.php>

有一系列內容完整、提供IPv6基本概念的介紹文章。有關主題的最佳入門。

<http://www.bieringer.de/linux/IPv6/>

在此處可找到Linux IPv6-HOWTO和許多與此主題相關的連結。

<http://www.6bone.net/> 如果想要加入發掘IPv6網路的行列，請瀏覽此網站。

<http://www.ipv6.org/> 所有有關IPv6的入門資訊。

RFC 2640 有關IPv6的基本RFC。

IPv6 Essentials 描述此主題所有重要面向的書籍。Silvia Hagen:*IPv6 Essentials*. O'Reilly & Associates, 2002 (ISBN 0-596-00125-8)。

22.3 名稱解析

DNS協助指定IP位址給一或多個名稱以及指定名稱給IP位址。在Linux中，這種轉換通常是由已知為`bind`的特殊類型軟體執行的。處理這個轉換的機器稱

為「名稱伺服器」(*name server*)。名稱組成階層系統，其中每個名稱元件以點分隔。但是，名稱階層與上述的IP位址階層無關。

考慮使用完整名稱，如`laurent.suse.de`，以`hostname.domain`格式來表示。完整名稱，也就是「完整領域名稱」(*Fully Qualified Domain Name*, *FQDN*)，是由主機名稱和領域名稱(`suse.de`)組成的。後者也包含了「最上層領域」(*top level domain*)或TLD(`de`)。

TLD指定因為過去的緣故變得相當混淆。習慣上，美國使用三個字母的領域名稱。全世界的其他國家，則是使用兩個字母的ISO國際代碼為標準。除此之外，2000年引入了較長的TLD，代表特定活動範圍(例如，`.info`、`.name`、`.museum`)。

在早期的網際網路(1990年前)，是使用檔案`/etc/hosts`儲存網際網路上所有機器的代表名稱。這種方式，對於連接到網際網路連線、快速增長的電腦數量層面而言，很快就證實是不切實際的。基於此因素，又開發出分散式的資料庫，以廣泛分散的方式來儲存主機名稱。這種資料庫與名稱伺服器類似，沒有有關網際網路上所有主機的立即可用資料，但是可以分散要求到其他名稱伺服器。

階層的最上層是由「root名稱伺服器」(*root name server*)所使用。這些root名稱伺服器管理最上層領域，且由「網路資訊中心」(*Network Information Center*, *NIC*)負責管理。每個root名稱伺服器知道負責指定最上層領域的名稱伺服器。有關最上層領域NIC的資訊可從<http://www.internic.net>取得。

DNS的功能不只是解析主機名稱。名稱伺服器也知道哪個主機，即「郵件交換器」(*Mail Exchanger*, *MX*)，負責接收該領域的電子郵件。

若要讓您的機器能夠解析IP位址，它必須知道至少一個名稱伺服器及其IP位址。透過YaST的幫助可以輕鬆指定這類名稱伺服器。如果您使用數據機撥接連線，完全不需要手動設定名稱伺服器。撥接通訊協定在連線建立時會提供名稱伺服器位址。關於使用SUSE LINUX存取名稱伺服器的組態，在章24, 領域名稱系統中有詳細的說明。

`whois`通訊協定與DNS密切相關。利用此程式，可快速找出伺服器負責哪個指定領域。

22.4 使用YaST手動設定網路連線

電腦中必須有支援的網路卡。通常，在安裝時會偵測網路卡，並且會載入適當的驅動程式。如果要確定您的網路卡是否已經與適當的驅動程式正確整合，請輸入指令`ip address list eth0`。其輸出會列出有關`eth0`網路設備的所有資訊或顯示錯誤訊息。

如果是以模組的形式來執行網路卡的核心支援(在SUSE 核心中為預設)，則必須在 `/etc/sysconfig/hardware/hwcfg-*` 中輸入模組名稱。如果未指定，則 `hotplug` 會自動選擇一個驅動程式。不論是哪一種類型的網路卡(熱插式或內建)，`hotplug` 都會為其指定驅動程式。

22.4.1 使用YaST 設定網路卡

啓動模組後，YaST 會顯示一般網路組態對話方塊。上半部顯示一個清單，列出所有尚未設定的網路卡。開機時自動偵測到的網路卡名稱會列在清單中。無法偵測的設備則列為‘其他(未偵測到)’。對話方塊下半部顯示目前已設定組態的設備清單，與其網路類型和位址。您現在可以設定新的網路卡或變更現有的組態設定。

手動設定網路卡

無法自動偵測到的網路卡的組態設定(列為‘其他’的網路卡) 包括下列項目：

網路組態 設定介面的設備類型和組態名稱。從提供的選項選取設備類型。根據您的需要指定一個組態名稱。預設值通常相當有用，而且是可接受的。有關組態名稱命名慣例的資訊，請參閱 `getcfg` 的手冊頁。

核心模組 ‘硬體組態名稱’ 會指出 `/etc/sysconfig/hardware/hwcfg-*` 檔案的名稱，其中包含網路卡的硬體設定，例如，合適的核心模組名稱。YaST 通常會建議有用的 PCMCIA 和 USB 硬體名稱。對於其他硬體來說，只有在網路卡使用 `hwcfg-static-0` 設定時，0 才有意義。

如果網路卡是 PCMCIA 或 USB 設備，請啓用個別的核取方塊並使用‘下一步’來結束對話方塊。如果不是，請由‘從清單選取’選擇您的網路卡模組。YaST 會自動選取合適的核心模組。使用‘下一步’來結束此對話方塊。

設定網路位址

設定介面的設備類型和組態名稱。從提供的選項選取設備類型。根據您的需要指定一個組態名稱。預設值通常相當有用，而且是可接受的。有關組態名稱命名慣例的資訊，請參閱 `getcfg` 的手冊頁。

如果您選取‘無線’做為介面的設備類型，請設定操作模式、網路名稱(ESSID) 以及下一個對話方塊‘無線網路卡組態’中的加密。按一下‘確定’完成您網路卡的組態設定。請參閱節 17.1.3, “使用 YaST 進行設定” 中有關無線網路卡組態的詳細資訊。如果要設定其他介面類型，請接著進行網路位址設定：

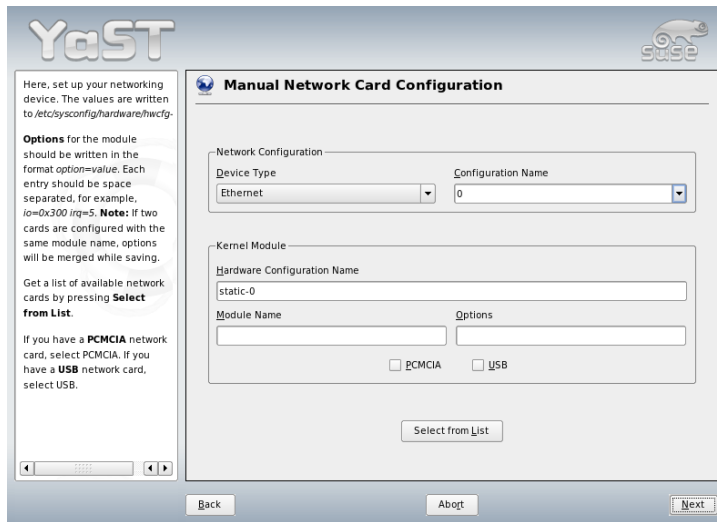


Figure 22.3: 網路卡的組態

‘自動位址設定(透過DHCP)’ 如果您的網路包含DHCP 伺服器，您可以用它來自動設定您的網路位址。如果您用的是DSL 連線而非ISP 指定的靜態IP 時，應該使用此選項。如果您決定使用DHCP，請在選取‘DHCP 用戶端選項’之後設定其詳細資料。指定DHCP 伺服器是否應該執行廣播要求以及使用任何識別碼。根據預設值，DHCP 伺服器使用網路卡的硬體位址來識別介面。如果您已設定了虛擬主機，讓不同的主機透過相同介面來通訊，則必須以識別碼來辨別彼此。

‘設定靜態位址’ 如果您有靜態位址，請啓用對應的核取方塊。然後輸入您網路的位址和子網路遮罩。預設的子網路遮罩應該會符合一般家用網路的需求。

選取‘下一步’來結束對話方塊，或接著設定主機名稱、名稱伺服器 and 路由詳細資料(請參閱和)。

‘進階’可讓您指定更複雜的設定。在‘詳細資料’下，可以使用‘使用者控制’，將管理員(root)的網路卡控制權委託給一般使用者。就行動操作而言，可讓使用者更有彈性地調整變更的網路連線，因此可以自行控制介面的啓用和停用。可在此對話方塊中設定MTU (傳輸單元最大值) 和‘設備啓用’類型。

22.4.2 數據機

在「YaST 控制中心」，可在「網路卡設備」下存取數據機組態。如果無法自動偵測到您的數據機，請開啓手動組態對話方塊。在開啓的對話方塊中，輸入數據機在「數據機設備」下用來連線的介面。

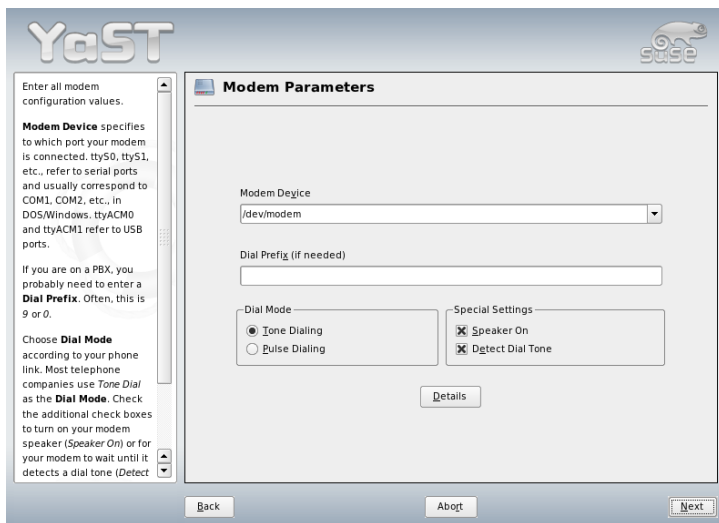


Figure 22.4: 數據機組態

如果您使用的是專用交換機(PBX)，您可能需要輸入撥號前置號碼。通常為0。請參閱PBX 隨附的說明。另外，請選擇是否使用按鍵式或轉盤式撥號、是否應該打開喇叭以及數據機偵測撥號音前是否應等待。如果數據機連接到分機，就不能啓用最後一個選項。

在「詳細資料」下，設定傳輸速率和數據機的啓始字串。只有在無法自動偵測您的數據機或是數據機需要特殊設定來傳輸資料時，才能變更這些設定。這些設定主要適用於ISDN 終端機介面卡。按一下「確定」來結束此對話方塊。如果要在沒有root 許可權的情況下將數據機控制權委託給一般使用者，請啓用「使用者控制」。使用此方式，使用者即可啓用和停用介面，而不需要管理員的許可。在「撥號前置號碼regex」下，指定一個一般表示式。KInternet 中的「撥號前置號碼」（一般使用者均可修改），必須符合此一般表示式。如果此欄位為空白，則使用者必須有管理員許可才能設定不同的「撥號前置號碼」。

在下一個對話方塊中，選擇ISP (網際網路服務提供者)。如果要從國內的ISP 預先定義清單中選擇，請選取‘國家’。或者，可按一下‘新增’來開啓一個對話方塊，您可以在其中輸入您的ISP 資料。這包括撥接連線名稱、ISP 名稱以及您的ISP 提供的登入名稱和密碼。啓用‘永遠詢問密碼’，以提示您在每次連線時輸入密碼。

在最後一個對話方塊中，可以指定其他的連線選項：

‘視需要撥號’ 如果您啓用了視需要撥號，請至少指定一個名稱伺服器。

‘連線時修改DNS’ 此核取方塊是依照預設啓用的，每次您連線到網際網路時，即會更新名稱伺服器位址。不過，如果您啓用了‘視需要撥號’，則會停用此選項並提供一個固定的名稱伺服器位址。

‘自動取回DNS’ 如果提供者沒有在連線後傳輸其領域名稱伺服器，則應停用此選項並手動輸入DNS 資料。

‘簡易模式’ 此選項預設為啓用。使用它，就會忽略ISP 伺服器送出的輸入提示，以避免干擾連線過程。

‘啓動防火牆’ 選取此選項即可啓用SUSE 防火牆，在您連線到網際網路時提供保護，防止外來攻擊。

‘閒置逾時(秒)’ 使用此選項來指定網路靜止一段時間後即自動中斷數據機的連線。

‘IP 詳細資料’ 這會開啓位址組態對話方塊。如果您的ISP 沒有為您的主機指定動態IP 位址，請停用‘動態IP位址’然後輸入您主機的本地IP 位址和遠端IP 位址。請向您的ISP 詢問此資訊。讓‘預設路由’保持為啓用狀態並選取‘確定’來結束此對話方塊。

選取‘下一步’回到原先的顯示數據機組態摘要的對話方塊。按一下‘完成’以結束此對話方塊。

22.4.3 ISDN

使用此模組來為您的系統設定一個或多個ISDN 卡。如果YaST 無法自動偵測您的ISDN 卡，請以手動選取它。可以有多個介面，但多個ISP 只能設定一個介面。在接下來的對話方塊中，設定網路卡正常運作所需的ISDN 選項。

在下一個對話方塊中(如圖形22.5, “ISDN 組態” 中所示)，選取要使用的通訊協定。預設為‘Euro-ISDN (EDSS1)’，但如果是較舊或較大型的交換機，則選取‘1TR6’。如果您是在美國，請選取‘NI1’。在相關欄位中選取您的國家。其對



Figure 22.5: ISDN 組態

應的國家代碼會顯示在旁邊的欄位。最後，提供您的‘區域號碼’和撥號前置號碼(如有需要)。

‘啓動模式’定義應該如何啓動ISDN 介面：‘OnBoot’會使系統每次啓動時都起始ISDN 驅動程式。‘手動’會要求您使用`rcisdn start` 指令來載入ISDN 驅動程式以做為`root`。用於PCMCIA 或USB 設備的‘熱插式設備’，會在插入設備後載入驅動程式。完成上述所有設定之後，請選取‘確定’。

在下一個對話方塊中，指定您ISDN 卡的介面類型，並將ISP 新增至現有的介面。介面可能為SyncPPP 或RawIP 類型，但大多數ISP 是在SyncPPP 模式中操作，其說明如下。

視您的特殊設定而定，在‘我的電話號碼’中必須輸入的號碼會有所不同：

ISDN 卡直接連至電話插孔 標準ISDN 連線會提供3 組電話號碼(稱為多重用戶號碼或MSN)。如果用戶要求更多號碼，最多可有10 組。必須在此輸入其中一組MSN，但不需要您的區域號碼。如果您輸入錯誤的號碼，您的電話操作員會自動回復指定給您的ISDN 連線的第一組MSN。

連接至電話交換機的ISDN 卡 此外，視安裝的設備而定，組態可能會有所不同：

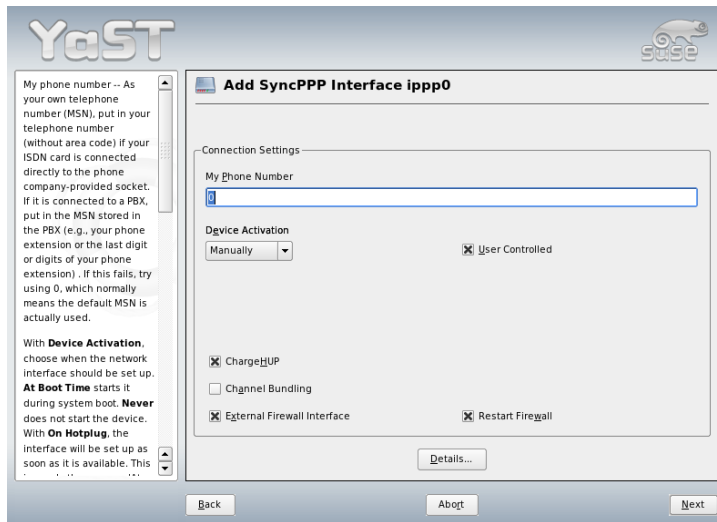


Figure 22.6: ISDN 介面組態

1. 小型的家用電話交換機大多使用Euro-ISDN (EDSS1) 通訊協定來撥接內線電話。這些電話交換機有一個內部S0 匯流排並在連線設備上使用內部號碼。
使用其中一組內部號碼做為您的MSN。您至少可使用一組電話交換機的MSN，這些MSN已啟用可直接對外撥號。如果無法使用，請嘗試撥0。如需更多資訊，請參閱您的電話交換機隨附的文件。
2. 較大型的商用電話交換機通常使用1TR6 通訊協定來撥接內線電話。其MSN稱為EAZ，通常可對應直撥號碼。如果要在Linux中設定組態，請輸入EAZ的最後一碼即可。最後一步是試著撥從1到9的每個數字。

如果想在下一個收費單位開始之前終止連線，可啟用‘ChargeHUP’。不過，要記得並不是每個ISP都適用。您也可以勾選對應的核取方塊來啟用通道合併(多重連結PPP)。最後，您可以選取‘啓動防火牆’來為您的連線啓用SuSEfirewall2。如果要讓一般使用者不需要管理員許可即可啓用和停用介面，請選取‘使用者控制’。

‘詳細資料’會開啓一個對話方塊，用來執行更複雜的連線配置，與一般的家庭使用者不相關。選取‘下一步’來結束此對話方塊。

下一個對話方塊中可進行IP位址設定。如果您的提供者沒有給您一個靜態IP，請選取‘動態IP位址’。或者，可根據您ISP的規格，在提供的欄位中輸入您主機的本地IP位址和遠端IP位址。如果該介面應做為網際網路的預設路由，請選取‘預設路由’。每個主機只能設定一個介面做為預設路由。選取‘下一步’來結束此對話方塊。

下列對話方塊中可讓您設定您的國家和選取ISP。清單中所列的ISP只有撥號計費提供者。如果您的ISP不在清單上，請選取‘新增’。會開啓‘提供者參數’對話方塊，您可在其中輸入您ISP的所有詳細資料。輸入電話號碼時，在數字之間不能有空白或逗號。最後，請輸入您的ISP所提供的登入名稱和密碼。完成後，請選取‘下一步’。

如果要在獨立的工作站上使用‘視要求撥號’，請指定名稱伺服器(DNS伺服器)。大多數ISP均支援動態DNS，即在您每次連線時ISP都會送出一個名稱伺服器的IP位址。不過，如果您使用的是單一工作站，您仍需提供一個替代位址，例如192.168.22.99。如果您的ISP不支援動態DNS，請指定ISP的名稱伺服器IP位址。如有需要，可為連線指定一個時限—未使用網路的時間(以秒為單位)，之後即會自動終止連線。使用‘下一步’來確認您的設定。YaST會顯示已設定介面的摘要。如果要啓用這些設定，請選取‘完成’。

22.4.4 纜線數據機

在某些國家(奧地利、美國)，透過電視纜線來存取網際網路是相當普遍的。有線電視用戶通常會有一部數據機，一邊連接到電視纜線的輸出端子，另一邊(使用10Base-TG雙絞纜線)連接到電腦網路卡。接著纜線數據機會以固定的IP位址提供專用的網際網路連線。

視您的ISP所提供的說明而定，設定網路卡時可選取‘自動位址設定(透過DHCP)’或‘靜態位址設定’。現在大多數提供者都使用DHCP。靜態IP位址通常是特殊用戶帳戶的一部分。

22.4.5 DSL

如果要設定您的DSL設備，請從YaST‘網路卡設備’區段選取‘DSL’模組。此YaST模組含有多個對話方塊，可根據下列其中一種通訊協定在其中設定DSL連線的參數：

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)

- CAPI for ADSL (Fritz 網路卡)
- 點對點通道通訊協定(PPTP) — 奧地利

PPPoE 或 PPTP 類型的DSL 連線組態，會要求對應的網路卡必須正確設定。如果您尚未完成此步驟，請先選取‘設定網路卡’來進行設定(請參閱節22.4.1, “使用YaST 設定網路卡”)。如果是DSL 連線，會自動指定位址但不是透過DHCP，，因此您不可啓用‘自動位址設定(透過DHCP)’ 選項，而是應該輸入介面的靜態虛擬位址，例如192.168.22.1。在‘子網路遮罩’中，輸入255.255.255.0。如果您要設定獨立的工作站，則應保留‘預設閘道’為空白。

‘IP 位址’和‘子網路遮罩’中的值只是預留位置。只用來啓始網路卡，而不代表DSL 連線等等。

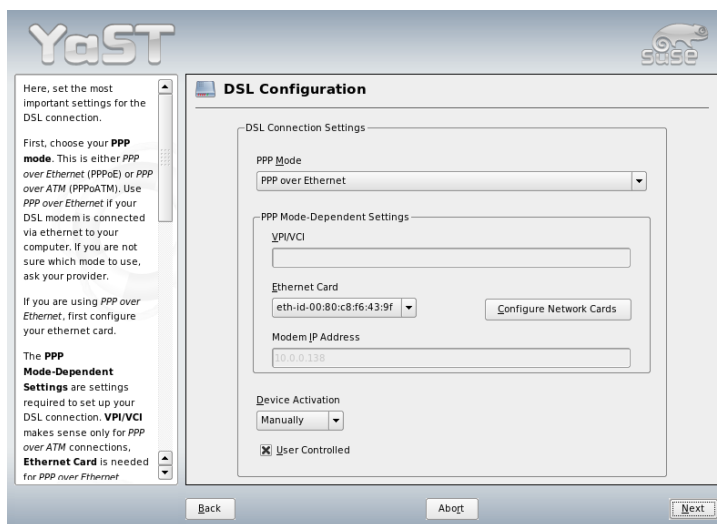


Figure 22.7: DSL 組態

如果要開始設定DSL 組態(請參閱圖形22.7, “DSL 組態”), 請先選取PPP 模式和用來連接DSL 數據機的乙太網路卡(大多為eth0)。然後使用‘設備啓用’來指定是否

要在開機時建立DSL連線。按一下‘使用者控制’來授權一般使用者，讓他們不需要root許可權即可透過KInternet來啓用或停用介面。您也可以使用該對話方塊來選擇您的國家和當地一些主控它的ISP。後續的DSL組態對話方塊詳細資料會視目前已設定的選項而定，因此下列段落只提供簡單的介紹。如需可用選項的詳細資訊，請參閱對話方塊中的詳細說明。

如果要在獨立的工作站上使用‘視要求撥號’，請指定名稱伺服器(DNS伺服器)。大部份ISP均支援動態DNS—每次連線時，名稱伺服器送出的IP位址。不過，如果是單一工作站，請提供一個替代位址，例如192.168.22.99。如果您的ISP不支援動態DNS，請輸入您ISP提供的名稱伺服器IP位址。

‘閒置逾時(秒)’定義網路靜止一段時間後即自動中斷連線。合理的逾時秒數為60至300秒之間。如果停用‘視需要撥號’，將逾時值設為零將有助於避免自動掛斷。

T-DSL的組態非常類似DSL的設定。只要選取‘T-Online’做為您的提供者，YaST就會開啓T-DSL組態對話方塊。在此對話方塊中，提供一些T-DSL要求的其他資訊—線路ID、T-Online號碼、使用者代碼和您的密碼。在您加入T-DSL後，應該會收到這些資訊。

22.5 手動設定網路連線

網路軟體的手動組態應該永遠是最後的替代方案。建議使用YaST。不過，這個有關網路組態的背景資訊也可協助您使用YaST。

所有內建的網路卡以及熱插式網路卡(PCMCIA、USB、一些PCI卡)會透過熱插方式進行偵測到並加以設定。系統是透過下列兩種不同的方法來偵測網路卡：第一種是以實體設備，第二種則是以介面。插入設備或偵測到設備時則會觸發熱插事件。此熱插事件使用程序檔/sbin/hwup觸發設備的起始化。起始化網路卡作為新的網路介面時，核心會產生另一個熱插事件，使用/sbin/ifup觸發介面的設定。

核心根據介面名稱註冊的暫時順序加以編號。起始化順序是指定名稱的決定因素。如果數個網路卡中的一個失敗，所有依序起始化網路卡的編號就會改變。對於真正的可熱插式網路卡，設備連接的順序才是決定因素。

為了讓組態具有彈性，已經分開設備(硬體)及介面的組態，而且不再以介面名稱來管理組態與設備和介面之間的對應。設備組態位於/etc/sysconfig/hardware/hwcfg-*。介面組態位於/etc/sysconfig/network/ifcfg-*。組態名稱是以描述設備及其關聯介面的方式來指定的。因為之前驅動程式與介面名稱的對應需要靜態介面名稱，所以此對應不再發生於/etc/modprobe.conf。在這種新的概念中，此檔案中的別名項目造成不想要的副作用。

組態名稱(`hwcfg-` 或 `ifcfg-` 之後的任何項目)都可以透過插槽、裝置專用ID或介面名稱來描述設備。例如，PCI卡的組態名稱可以是`bus-pci-0000:02:01.0` (PCI插槽) 或 `vpid-0x8086-0x1014-0x0549` (廠商和產品ID)。相關聯介面的名稱可以是`bus-pci-0000:02:01.0` 或 `wlan-id-00:05:4e:42:31:7a` (MAC位址)。

若要指定特定網路組態到任何特定類型的卡(一次僅能插入一種)而不是特定卡，請選取較不特定的組態名稱。例如，`bus-pcmcia` 可以用於所有PCMCIA卡。在另一方面，之前的介面類型會限制名稱。例如，`wlan-bus-usb` 可以指定給連接到USB埠的WLAN卡。

系統永遠使用最佳描述介面或提供介面之設備的組態。搜尋最適用組態是由`/sbin/getcfg`處理的。`getcfg`的輸出會傳達可以用來描述設備的所有資訊。有關組態名稱規格的詳細資料，請參閱`getcfg`的手冊頁。

透過描述的方法，即使網路設備不一定永遠以相同的順序起始化，網路介面還是可以具有正確的組態設定。不過，介面的名稱仍然取決於起始化順序。有兩種方式可以確保確實存取特定網路卡介面：

- `/sbin/getcfg-interface<configuration name>` 傳回相關聯網路介面的名稱。因此，可以輸入一部份的組態名稱(例如，防火牆、`dhcpd`、路由、各種虛擬網路介面(通道))，但不是所有的組態檔案都可以取代介面名稱，因為這種名稱並不是永久不變的。
- 永久介面名稱可以指定給其組態不具有介面名稱的所有介面。透過介面組態(`ifcfg-*`)中的`PERSISTENT_NAME=<pname>`項目可以執行這個動作。然而，永久名稱(`pname`)不能與核心自動指定的名稱相同。所以，不允許`eth*`、`tr*`、`wlan*`、`qeth*`、`iucv*`等名稱。相反地，請使用`net*`或是`external`、`internal`、`dmz`之類的描述名稱。永久名稱僅能在註冊後立即指定給介面，這表示必須重新載入網路卡的驅動程式或執行`hwup <device description>`。針對此用途，僅使用指令`rcnetworkrestart`是不夠的。

使用永久介面名稱

永久介面名稱的使用尚未測試於所有方面。因此，一些應用程式可能無法自由地處理選取的介面名稱。如果您碰到這類問題，請透過<http://www.suse.de/feedback>通知我們。

`ifup`需要現有介面，因為它不會起始化硬體。硬體的起始化是由指令`hwup` (由`hotplug`或`coldplug`執行)處理的。起始化設備時，會透過`hotplug`自動

為新介面執行 `ifup`，如果開始模式是 `onboot`、`hotplug` 或 `auto`，就會設定介面，而且會啓動 `network` 服務。之前，是使用指令 `ifup <interfacename>` 來觸發硬體初始化。現在已經反轉此程序。首先，會初始化硬體元件，接著是所有其他的動作。採取這種方式，則永遠可以使用現有的組態集，儘可能以最佳的方式設定不同數目的設備。

表格 22.5, “手動網路組態程序檔” 總結了與網路組態相關的最重要程序檔。只要可能，會依硬體和介面區分程序檔。

Table 22.5: 手動網路組態程序檔

組態階段	指令	功能
硬體	<code>hw{up,down,status}</code>	<code>hw*</code> 程序檔由熱插式子系統執行，用來初始化設備、復原初始化、或查詢設備狀態。詳細資訊請參閱 <code>hwup</code> 的手冊頁。
介面	<code>getcfg</code>	<code>getcfg</code> 可以用來查詢與組態名稱或硬體描述相關聯的介面名稱。詳細資訊請參閱 <code>getcfg</code> 的手冊頁。
介面	<code>if{up,down,status}</code>	<code>if*</code> 程序檔會開啓現有網路介面或傳回指定介面的狀態。詳細資訊請參閱 <code>ifup</code> 的手冊頁。

有關熱插拔和永久設備名稱的詳細資訊，請參閱章 18, 熱插拔系統與章 19, 使用 `udev` 的動態裝置節點。

22.5.1 組態檔

本節提供網路組態檔的概觀，並說明其用途和使用的格式。

`/etc/syconfig/hardware/hwcfg-*`

這些檔案包含網路卡及其他設備的硬體組態；其中包含所需的參數，如核心模組、啓動模式和程序檔關聯。詳細資訊請參閱 `hwup` 的手冊頁。無論現有硬體為何，啓動 `coldplug` 時會套用 `hwcfg-static-*` 組態。

/etc/sysconfig/network/ifcfg-*

這些檔案包含網路介面的組態；其中包含啓動模式和IP位址等資訊。可以使用的參數請參閱ifup的手冊頁。此外，如果一般設定僅用於一個介面，則檔案dhcp、wireless和config中的所有變數都可以用於ifcfg-*檔案。

/etc/sysconfig/network/config、dhcp、wireless

檔案config包含ifup、ifdown、和ifstatus行爲的一般設定；dhcp則包含無線區域網路介面卡之DHCP和wireless的設定。所有三個組態檔中的變數都已註解，而且可以在ifcfg-*檔案中使用，以更高優先順序處理。

/etc/sysconfig/network/routes,ifroute-*

TCP/IP封包的靜態路由在此決定。各種系統工作所需的所有靜態路由可以在/etc/sysconfig/network/routes檔案中輸入：路由至主機、透過閘道的路由至主機，及路由至網路。針對需要個別路由的每個介面，定義其他的組態檔：/etc/sysconfig/network/ifroute-*。以介面的名稱取代*。在路由組態檔中的項目看起來就像這樣：

```
DESTINATION          GATEWAY NETMASK  INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION          GATEWAY PREFIXLEN INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION/PREFIXLEN GATEWAY -       INTERFACE [ TYPE ] [ OPTIONS ]
```

若要省略GATEWAY、NETMASK、PREFIXLEN或INTERFACE，請改以-替代。可以省略TYPE與OPTIONS項目。

路由的目的地是在第一個資料欄。這個資料欄可能包含網路或主機的IP位址，這是指「可到達的」名稱伺服器、完整合格的網路或主機名稱。

第二個資料欄包含預設的閘道或是可以存取主機或網路的閘道。第三個資料欄包含在閘道後面的網路或主機的網路遮罩。例如，遮罩為255.255.255.255，供在閘道後面的主機使用。

最後一欄只與連線至本地主機的網路相關，例如迴路、乙太網路、ISDN、PPP以及虛擬設備。必須在這裏輸入設備名稱。

/etc/resolv.conf

主機所屬的領域指定於此檔案(關鍵字search)；另外也會列出要存取的名稱伺服器位址的狀態(關鍵字nameserver)。可以指定多個領域名稱。解析不完整的名稱時，會嘗試附加個別search項目產生一個名稱。透過輸入數行且每行都以nameserver開頭的方法，可使用多個名稱伺服器。在註解前加上#符號。YaST在此檔案中輸入指定名稱伺服器。範例22.5，“/etc/resolv.conf”顯示出/etc/resolv.conf有可能的樣子。

Example 22.5: /etc/resolv.conf

```
# Our domain
search example.com
#
# We use sun (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

一些如pppd(wvdial)、ippdd(isdn)、dhcp(dhcpd和dhclient)、pcmcia及hotplug之類的服務會修改檔案/etc/resolv.conf，方法是使用程序檔modify_resolvconf。如果檔案/etc/resolv.conf已經由此程序檔暫時修改，它會包含預先定義的註解，提供的資訊包含修改它的服務、原始檔案的備份位置以及如何關閉自動修改機制。如果數次修改/etc/resolv.conf，檔案會以巢狀形式包含所作的修改。即使反轉時使用與修改順序不同的順序，還是可以完全反轉此程序。需要這種彈性的服務包括isdn、pcmcia、hotplug。

如果沒有以正常的方式終止服務，可以使用modify_resolvconf還原原始檔。另外，在系統開機時，會執行檢查，看看是否有沒有清理、已修改的resolv.conf(例如，系統當機後)，在這種情況下，會還原原始(未修改)的resolv.conf。

YaST使用指令modify_resolvconf check找出是否已修改resolv.conf，接著將警告使用者還原檔案會遺失所有變更。除此之外，YaST不會依靠modify_resolvconf，意即會透過YaST變更resolv.conf的影響與手動變更的影響是一樣的。無論是哪種情形，變更永遠是有效的。上述提及的服務所需的修改只是暫時的。

/etc/hosts

在此檔中(請參閱範例22.6, "/etc/hosts")，IP位址是指定給主機名稱。如果沒有執行任何名稱伺服器，將使用此IP連線設定的所有主機將列示於此。對於每個主機，分別在檔案中輸入一行包含IP位址、完整主機名稱及主機名稱的項目。IP位址必須在行的開頭，然後以空格和定位點分隔這些項目。註解的前面永遠是#符號。

Example 22.6: /etc/hosts

```
127.0.0.1 localhost
192.168.0.20 sun.example.com sun
192.168.0.1 earth.example.com earth
```

/etc/networks

在此檔中，網路名稱會轉換為網路位址。格式與hosts檔案格式相似，但是網路名稱在位址前面。請參閱範例22.7, “/etc/networks”。

Example 22.7: /etc/networks

```
loopback      127.0.0.0
localnet      192.168.0.0
```

/etc/host.conf

名稱解析，即透過解析程式庫 (Resolver Library) 翻譯主機及網路名稱，是由此檔案控制的。該檔案僅用於與libc4 或libc5 連結的程式。對於目前的glibc程式，請參閱/etc/nsswitch.conf 中的設定。每個參數必須永遠是獨立一行。註解的前面是# 符號。表格22.6, “/etc/host.conf 的參數” 顯示出可用的參數。/etc/host.conf 範例是顯示在範例22.8, “/etc/host.conf”。

Table 22.6: /etc/host.conf 的參數

順序主機、繫結	指定針對名稱解析存取服務的順序。可用的引數為(以空格或是逗號分隔)： <i>hosts</i> ：搜尋/etc/hosts 檔案 <i>bind</i> ：存取名稱伺服器 <i>NIS</i> ：使用NIS
<i>multi on/off</i>	定義在/etc/hosts 中所輸入的主機是否可以有多個IP 位址。
<i>nospoof on spoofalert on/off</i>	這些參數會影響名稱伺服器 <i>spoofing</i> ，但除此之外，並不會對網路組態有任何影響。
<i>trim domainname</i>	指定的領域名稱在主機名稱解析後會與主機名稱分隔(只要主機名稱包括領域名稱)。只有在本地領域分離出來的名稱位於/etc/hosts 檔案，但是仍然使用附加的領域名稱進行辨識時，這個選項才有用。

Example 22.8: /etc/host.conf

```
# We have named running
order hosts bind
# Allow multiple addrs
multi on
```

/etc/nsswitch.conf

GNU C Library 2.0 的介紹伴隨「名稱服務切換」(NSS, Name Service Switch)的介紹。詳細資訊請參閱man 5 nsswitch.conf 與GNU C Library 參考手冊。

查詢的順序定義於檔案/etc/nsswitch.conf。nsswitch.conf 範例是顯示在範例22.9, “/etc/nsswitch.conf”。註解從# 符號開始。在此範例中, hosts 資料庫下的項目表示要求是透過DNS 傳送到/etc/hosts (files) (請參閱章24, 領域名稱系統)。

Example 22.9: /etc/nsswitch.conf

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files

netgroup:    files
automount:   files nis
```

NSS 上可用的「資料庫」列示於表格22.7, “透過/etc/nsswitch.conf 的可用資料庫”。此外, 將來應該還有automount、bootparams、netmasks、和publickey。NSS 資料庫的組態選項是列在表格22.8, “NSS 「資料庫」的組態選項”。

Table 22.7: 透過/etc/nsswitch.conf 的可用資料庫

aliases	sendmail 所執行的郵件別名；請參閱man 5 aliases。
ethers	乙太網路位址。
群組	getgrent 所使用的使用者群組。請參閱group 的man 頁面。

主機	gethostbyname 及類似功能所使用的主機名稱與IP位址。
netgroup	在網路中有效的主機與使用者清單，以利控制存取權限，請參閱man 5 netgroup。
networks	getnetent 所使用的網路名稱與位址。
passwd	getpwent 所使用的使用者密碼；請參閱man 5 passwd。
protocols	getprotoent 所使用的網路通訊協定；請參閱man 5 protocols。
rpc	getrpcbyname 及類似功能所使用的遠端程序呼叫名稱與位址。
services	getservent 使用的網路服務。
shadow	getspnam 所使用的使用者遮蔽密碼；請參閱man 5 shadow。

Table 22.8: NSS 「資料庫」的組態選項

files	直接存取檔案，例如/etc/aliases
db	透過資料庫存取
nis、nisplus	NIS，請參閱章25, 使用NIS
dns	只能做為hosts 與networks 的延伸
compat	只能做為passwd、shadow 以及group 的延伸

/etc/nscd.conf

此檔案用來設定nscd(名稱服務快取精靈)。請參閱man 8 nscd 與man 5 nscd.conf。依預設，passwd 與groups 的系統項目是由nscd 快取。這對於目錄服務(如NIS 和LDAP) 的效能而言是很重要的，否則每次存取名稱或群組時都需要使用網路連線。預設是不會快取hosts，因為nscd 快取主機的機制會造成本地系統無法信任轉寄以及反向查詢檢查。不要要求nscd 快取名稱，而是設定快取DNS 伺服器。

如果啓用passwd 的快取，通常需要15 秒，才能辨識新增的本地使用者。使用指令rcnscd restart 重新啓動nscd，以縮短這段等待時間。

/etc/HOSTNAME

這是沒有附加領域名稱的主機名稱。機器開機時數個程序檔會讀取該檔案。它可以只包含一行，其中設定了主機名稱。

22.5.2 啓動程序檔

除了上述的組態檔之外，還有在機器開機時載入網路程式的各種程序檔。只要系統切換到其中一個 *multiuser runlevels*，就會啓動這些程序檔。在表格 22.9, “網路程式的一些啓動程序檔” 中提供了一些程序檔的說明。

Table 22.9: 網路程式的一些啓動程序檔

<code>/etc/init.d/network</code>	此程序檔會處理網路介面的組態。硬體必須先由 <code>/etc/init.d/coldplug</code> (透過 <code>hotplug</code>) 起始化。如果沒有啓動 <code>network</code> 服務，透過熱插拔插入網路介面時，將無法執行它們。
<code>/etc/init.d/inetd</code>	啓動 <code>xinetd</code> 。 <code>xinetd</code> 可用來使伺服器服務供系統使用。例如，只要開啓 FTP 連線，它即可啓動 <code>vsftpd</code> 。
<code>/etc/init.d/portmap</code>	啓動 RPC 伺服器所需的 <code>portmapper</code> ，例如 NFS 伺服器。
<code>/etc/init.d/nfsserver</code>	啓動 NFS 伺服器。
<code>/etc/init.d/sendmail</code>	控制傳送郵件的程序。
<code>/etc/init.d/ypserv</code>	啓動 NIS 伺服器。
<code>/etc/init.d/ypbind</code>	啓動 NIS 用戶端。

22.6 smpppd 做爲撥號助理

大部份的家用使用者都沒有連線到網際網路的專線，而是使用撥號連線。視撥號方法 (ISDN 或 DSL) 而定，連線是由 `ippdd` 或 `pppd` 來控制。基本上，連線所需做的事就是正確地啓動這些程式。

如果您有單一速率的連線，不會產生撥號連線的其他成本，請直接啓動個別的精靈。使用 KDE Applet 或指令行介面以控制撥號連線。如果網際網路閘道不是您

所使用的主機，您可能需要透過網路主機來控制撥號連線。

這裏將會需要smpppd。它會為輔助程式提供一致的介面並以兩個方向運作。首先，它會撰寫所需的pppd或ipppd程式，然後控制其撥號內容。其次，它會提供各個提供者給使用者程式，並傳輸關於連線目前狀態的資訊。因為smpppd也可以透過網路控制，所以它適合從私人子網路的工作站中，控制連至網際網路的撥號連線。

22.6.1 設定smpppd

smpppd所提供的連線會自動由YaST設定。實際的撥號程式kinternet與cinternet也是預先設定的。手動設定只需設定smpppd的其他功能，例如遠端控制。

smpppd的組態檔為/etc/smpppd.conf。根據預設，它不會啓用遠端控制。此組態檔最重要的選項為：

open-inet-socket = {yes—no} 若要透過網路控制smpppd，此選項必須設定為yes。smpppd所傾聽的埠為3185。如果此參數設為yes，則也應該設定bind-address、host-range以及password參數。

bind-address = {ip} 如果主機具有數個IP位址，請使用此參數以決定smpppd應該接受連線的IP位址。

host-range = {min ip} {max ip} host-range參數會定義網路範圍。在此範圍中的IP位址之主機擁有smpppd的存取權。所有不在此範圍中的主機都將拒絕存取。

password = {password} 透過指定密碼，將用戶端限制為授權的主機。因為這是純文字的密碼，您不應該高估它所提供的安全性。如果沒有指定密碼，則所有的用戶端都將允許存取smpppd。

slp-register = {yes—no} 使用此參數，smpppd服務就可以透過SLP在網路中宣告。

如需關於smpppd的詳細資訊，請參閱man 8 smpppd與man 5 smpppd.conf。

22.6.2 設定kinternet、cinternet 與qinternet 以供遠端使用

kinternet、cinternet 以及qinternet 可用以控制本地或遠端smpppd。cinternet 是圖形化kinternet 對等的指令行程式。qinternet 基本上與kinternet 相同，但是不使用KDE 程式庫，因此可以在沒有KDE 的情況下使用它，而且必須分開安裝。若要準備這些公用程式以供遠端smpppd 使用，請手動編輯/etc/smpppd-c.conf 組態檔或使用kinternet。這個檔案只使用三個選項：

sites = ;list of sites; 在此，可看出要搜尋smpppd 的前端。這些前端將會依這裏所指定的順序來測試選項。local 選項會命令建立連至本地smpppd 的連線。gateway 會指向閘道的smpppd。這個連線將會依config-file 中server 下的指定來建立。slp 會命令前端，連線至透過SLP 所找到的smpppd。

server = ;server; 此處指定smpppd 所執行的主機。

password = ;password; 插入為smpppd 所選取的密碼。

如果smpppd 為作用中，您現在可以嘗試存取它，例如，使用cinternet --verbose --interface-list。如果您在此時遭遇到困難，請參考man 5 smpppd-c.conf 以及man 8 cinternet

網路中的SLP服務

服務位置通訊協定 (SLP) 是開發用來簡化區域網路內的網路用戶端組態。若要設定網路用戶端(包含所有必要的服務)，一般而言管理員需要對於網路中可用的伺服器有詳細的瞭解。SLP 可讓區域網路中的所有用戶端，都知道某一服務的可用性。支援SLP 的應用程式可以使用散佈的資訊並可自動設定。

23.1	註冊自己的服務	362
23.2	在SUSE LINUX 中的SLP 前端	363
23.3	啓動SLP	363
23.4	若需詳細資訊	363

SUSE LINUX 支援使用透過SLP所提供的安裝來源以進行安裝，並包含許多具有SLP整合支援的系統服務。YaST與Konqueror兩者都有適當的SLP前端。您可以使用SLP以提供主要的功能給網路上的用戶端，例如安裝伺服器、YOU伺服器、檔案伺服器或是SUSE LINUX上的列印伺服器。

23.1 註冊自己的服務

許多在SUSE LINUX下的應用程式，透過libslp程式庫的使用，已經具有整合的SLP支援。如果尚未使用SLP支援來編譯服務，請使用下列其中一種方式讓SLP編譯服務：

透過/etc/slp.reg.d的靜態註冊 針對每個新的服務建立個別的註冊檔。下列是註冊掃描器服務的檔案範例：

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

在此檔案中最重要的一行為*service URL*，它是以service:開始。這包含服務類型(scanner.sane)以及位址(可在其中找到伺服器可用的服務)。*(\$HOSTNAME)*會以完整的主機名稱自動取代。接著可以找到相關服務的TCP埠名稱，它們之間是以冒號分隔。然後輸入服務應該出現的語言以及註冊期間的秒數。這些都應該使用逗號與服務URL分隔。在0與65535之間設定註冊期間的值。0會防止註冊。65535會移除所有的限制。

註冊檔另外也包含watch-tcp-port與description兩個變數。第一個變數將SLP服務宣告連結至是否為作用中的相關服務，因為slpd會檢查服務的狀態。第二個變數是對顯示在適當瀏覽器中的服務，提供更為精確的描述。

使用/etc/slp.reg的靜態註冊 上述程序的唯一差異是在中央檔案內群組所有的服務。

以slptool動態註冊 如果應該從專用的程序檔為SLP註冊服務，請使用slptool指令行前端。

23.2 在SUSE LINUX 中的SLP 前端

SUSE LINUX 包含數個前端，可透過網路的檢查及使用，來啓用SLP 資訊：

slptool slptool 是簡易的指令行程式，可以在網路中宣告SLP 查詢或宣告專用的服務。slptool --help 可以列出所有可用的選項與功能。也可以從處理SLP 資訊的程序檔呼叫slptool。

YaST SLP 瀏覽器 YaST 包含獨立的SLP 瀏覽器，以樹狀圖表在‘網路服務’→‘SLP 瀏覽器’下列出區域網路中所有透過SLP 所宣告的服務

Konqueror 當Konqueror 做為網路瀏覽器時，可以在slp:/ 顯示區域網路中所有可用的SLP 服務。在主要視窗中按一下圖示，以取得更多關於相關服務的詳細資訊。

如果您使用Konqueror 加上service:/，則請在瀏覽器視窗中按一次相關圖示，以設定與選取服務的連線。

23.3 啓動SLP

如果您要提供服務，必須在系統上執行slpd。如果只是要做服務查詢，並不需要啓動此精靈。就像大部份在SUSE LINUX 之下的系統服務一樣，slpd 精靈是利用獨立的init 程序檔來控制。預設精靈為非作用中。若要為持續某個工作階段而啓動它，請以root 的身份執行rcslpd start 以啓動它，並執行rcslpd stop 以停止它。請以restart 或status 執行重新啓動或狀態檢查。如果預設slpd 應該啓動，請以root 的身份執行insserv slpd 指令。這將會在系統開機時，自動在要啓動的服務集合中包括slpd。

23.4 若需詳細資訊

下列來源提供關於SLP 的進一步資訊：

RFC 2608、2609、2610 RFC 2608 一般會處理SLP 的定義。RFC 2609 會處理更加詳細的服務URL 語法，而RFC 2610 則會透過SLP 處理DHCP。

<http://www.openslp.com> OpenSLP 計劃的首頁。

file:/usr/share/doc/packages/openslp/*

這個目錄包含所有關於SLP的可用文件，包括README.SuSE (含有SUSE LINUX 詳細資訊)、上方所提及的RFC 以及兩個介紹性的HTML 文件。想要使用SLP 功能的程式設計人員，應該安裝openslp-devel 套件，以參閱它所提供的「程式設計人員指南」。

領域名稱系統

必須使用DNS (領域名稱系統) 將領域和主機名稱解析為IP 位址。例如，藉由這種方式，IP 位址192.168.0.1 會指定給主機名稱earth。在設定您自己的名稱伺服器前，請參閱節22.3, “名稱解析” 中有關DNS 的一般資訊。以下設定範例是指BIND。

24.1	使用YaST 進行設定	366
24.2	啓動名稱伺服器BIND	370
24.3	組態檔/etc/named.conf	374
24.4	區域檔案	378
24.5	區域資料的動態更新	381
24.6	安全交易	381
24.7	DNS 安全性	382
24.8	若需詳細資訊	383

24.1 使用YaST 進行設定

您可以使用YaST的DNS模組設定您區域網路的DNS伺服器。第一次啓動模組時，會啓動精靈以提示您決定有關伺服器管理的一些基本設定。完成此最初設定程序會產生一個非常基本的伺服器組態，應該在其基本方面都能運作。進階模式可以用來處理更進階的組態工作。

24.1.1 精靈組態

精靈包含三個步驟或對話方塊。在對話方塊中適當的地方，能夠讓您進入進階組態模式。

Forwarder 設定 第一次啓動模組時，請參閱圖形24.1, “DNS 伺服器安裝：Forwarder 設定” 中所顯示的對話方塊。它可以決定PPP精靈是否應該提供透過DSL或ISDN在撥接時forwarder的清單(‘PPP精靈設定Forwarder’)，或者是否要提供您自己的清單(‘手動設定Forwarder’)。

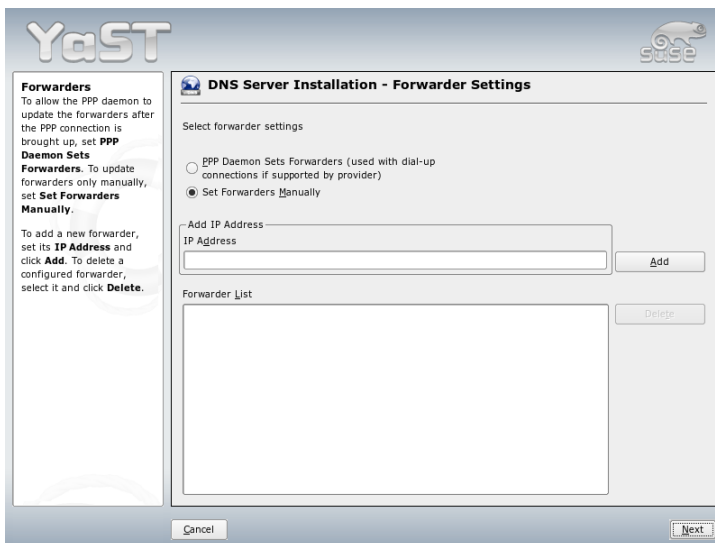


Figure 24.1: DNS 伺服器安裝：Forwarder 設定

DNS 區域 此對話方塊包含數個部分，負責管理區域檔案，在節24.4, “區域檔案” 中可找到它的相關說明。對於新區域，在‘區域名稱’ 中提供名稱。若要新增反向區域，名稱的結尾必須是.in-addr.arpa。最後，選取‘區域類型’ (主要或從屬)。請參閱圖形24.2, “DNS 伺服器安裝：DNS 區域”。按一下‘編輯區域’，設定現有區域的其他設定。若要移除區域，按一下‘刪除區域’。

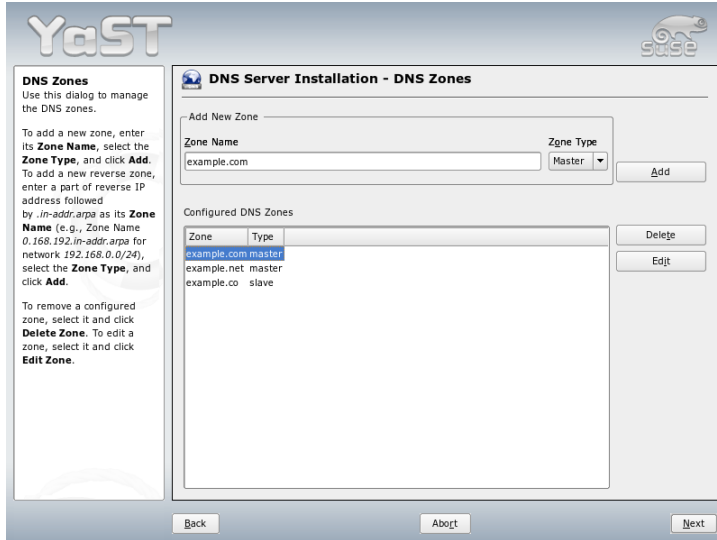


Figure 24.2: DNS 伺服器安裝：DNS 區域

完成精靈 在這個最後的對話方塊中，您可以開啓在安裝期間所啓動防火牆中的DNS 服務埠，並決定是否要啓動DNS。進階組態也可以從此對話方塊存取。請參閱圖形24.3, “DNS 伺服器安裝：完成精靈”。

24.1.2 進階組態

啓動模組後，YaST 會開啓顯示數個組態選項的視窗。完成該視窗可讓DNS 伺服器組態的基本功能就位運作：

啓動 在‘開機’下，定義DNS 伺服器預設該是‘開’或‘關’。若要立即啓動DNS 伺服器，選取‘立即啓動DNS 伺服器’；若要停止DNS 伺服器，選取‘立即

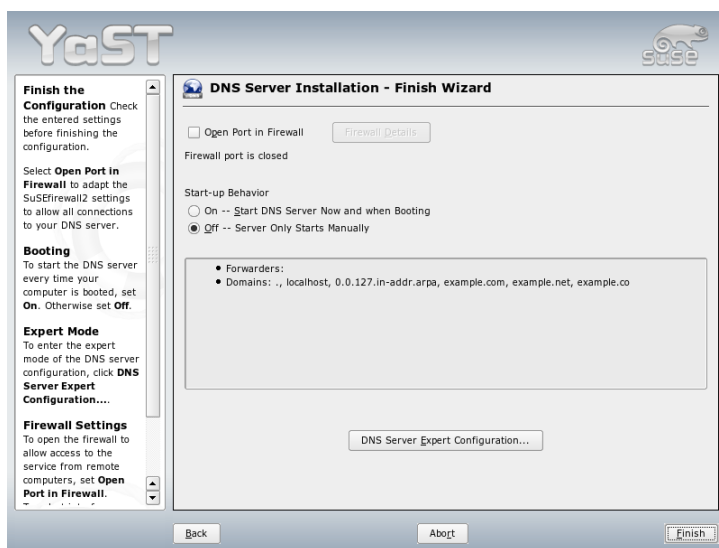


Figure 24.3: DNS 伺服器安裝：完成精靈

停止DNS 伺服器’。若要儲存目前的設定，選取‘立即儲存設定並重新啟動DNS 伺服器’。您可以使用‘在防火牆中開啓埠’在防火牆中開啓DNS 埠，並使用‘防火牆細節’修改防火牆設定。

Forwarder 這個對話方塊與啓動精靈組態後所開啓的對話方塊相同(請參閱)。

記錄 本區段允許您設定DNS 伺服器應該記錄的內容及方法。在‘記錄類型’下，指定DNS 伺服器應該寫入記錄資料的位置。選取‘記錄到系統記錄’來使用全系統的記錄檔 `/var/log/messages` 或選取‘記錄到檔案’指定不同的檔案。在後者情況下，另外也要指定最大檔案大小(以MB 為單位)以及要儲存的記錄檔數量。

進一步選項可從‘其他記錄’下存取：啓用‘記錄命名查詢’會記錄「每個」查詢，這個選項會讓記錄檔的大小變得非常大。所以，除了偵錯用途外，啓用此選項並不是理想的作法。若要記錄DHCP 與DNS 伺服器之間區域更新期間的資料流量，啓用‘記錄區域更新’。若要記錄從主伺服器到從屬伺服器區域傳輸期間的資料流量，啓用‘記錄區域傳輸’。請參閱圖形24.4, “DNS 伺服器：記錄”。

DNS 區域 這個對話方塊說明精靈的組態。請參閱節24.1.1, “精靈組態”。

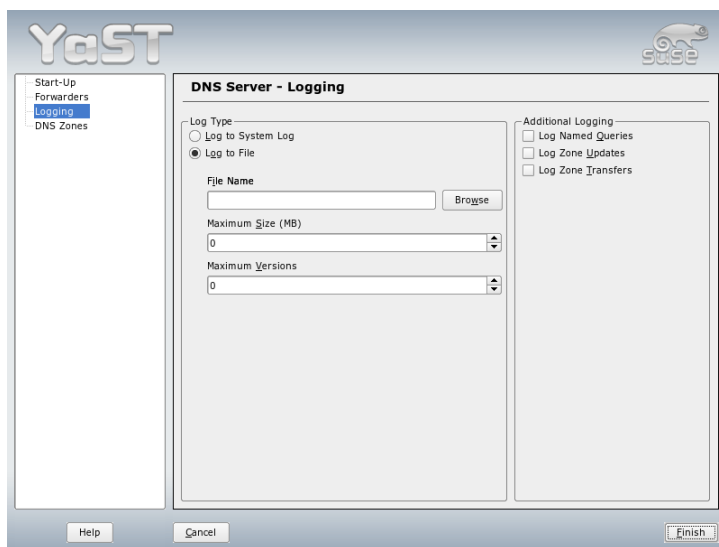


Figure 24.4: DNS 伺服器：記錄

從屬區域編輯器 如果在所述的步驟中選取的區域類型為‘從屬’，就會開啓此對話方塊。在‘主要DNS 伺服器’下，指定從屬伺服器將取得其資料的主要伺服器。若要限制對伺服器的存取，可以從清單選取其中一個ACL。請參閱圖形24.5, “DNS 伺服器：從屬區域編輯器”。

主要區域編輯器 如果在所述的步驟中選取的區域類型為‘主要’，就會開啓此對話方塊。對話方塊包含數頁：‘基本’ (先開啓的頁面) 、『NS 記錄’、‘MX 記錄’、‘SOA’ 以及 ‘記錄’。

區域編輯器(NS 記錄) 此對話方塊允許您定義指定區域的替代名稱伺服器。請確定您自己的名稱伺服器包含於清單中。若要新增記錄，在‘要新增的名稱伺服器’下輸入其名稱，然後使用‘新增’ 確認動作。請參閱圖形24.7, “DNS 伺服器：區域編輯器(NS 記錄)”。

區域編輯器(MX 記錄) 若要將目前區域的郵件伺服器新增到現有清單，請輸入對應位址及優先順序值。完成後，選取‘新增’ 確認該動作。請參閱圖形24.8, “DNS 伺服器：區域編輯器(MX 記錄)”。

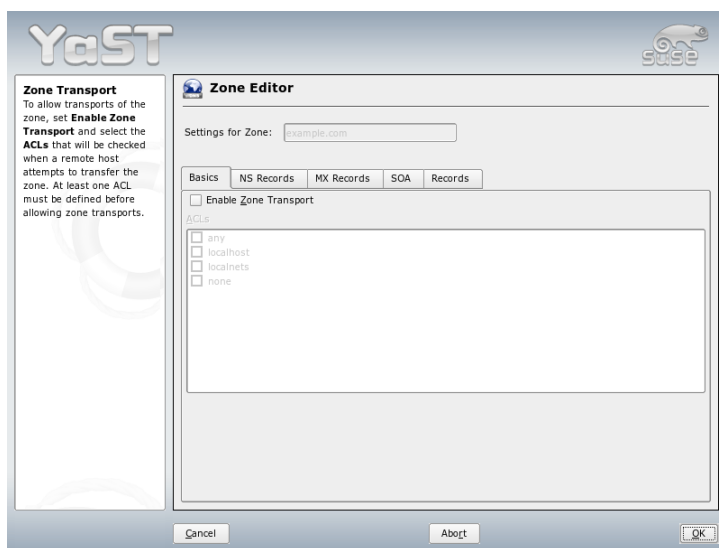


Figure 24.5: DNS 伺服器：從屬區域編輯器

區域編輯器(SOA) 此頁允許您建立SOA (授權開始) 記錄。如需個別選項的說明，請參閱範例24.6, “檔案/var/lib/named/world.zone”。透過LDAP 管理的動態區域，並不支援變更SOA 記錄。

區域編輯器(記錄) 這個對話方塊是管理名稱解析。在‘記錄金鑰’中，輸入主機名稱，然後選取其類型。‘A-Record’代表主項目。這個的值應該是IP 位址。‘CNAME’是別名。使用‘NS’與‘MX’類型，以取得‘NS 記錄’與‘MX 記錄’標籤中所提供的擴充資訊的詳細或部份記錄。這三個類型可以解析成現有的A 記錄。‘PTR’是供反向區域所使用。它是A 記錄的相反。

24.2 啓動名稱伺服器BIND

在SUSE LINUX 系統上，名稱伺服器BIND (*Berkeley Internet name domain*，柏克萊網際網路名稱領域) 已經預先設定，所以安裝後即可啓動，不會有任何問題。如果您已具有可以運作的網際網路連線，而且也在/etc/resolv.conf 中輸入了127.0.0.1 做為localhost 的名稱伺服器位址，表示您通常已經有工作

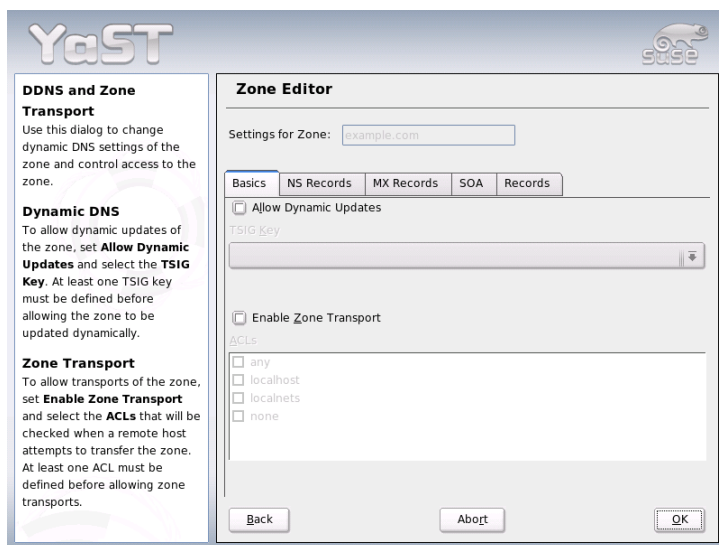


Figure 24.6: DNS 伺服器：區域編輯器(基本)

名稱解析，就不需要知道提供者的DNS。BIND 透過根名稱伺服器完成名稱解析，顯見處理程序較慢。一般而言，提供者的DNS 應該在 `forwarders` 之下的組態檔 `/etc/named.conf` 中輸入其IP 位址，以確保有效及安全的名稱解析。如果目前都能運作，名稱伺服器會當作純粹的「僅快取」名稱伺服器執行。只有您設定它自己的區域時，它才會變成適當的DNS。關於這個的簡單範例包含於 `/usr/share/doc/packages/bind/sample-config` 的文件中。

自動使用名稱伺服器資訊

根據網際網路連線或網路連線的類型，可以自動對目前的情況使用名稱伺服器資訊。若要這樣做，將檔案 `/etc/sysconfig/network/config` 中的變數 `MODIFY_NAMED_CONF_DYNAMICALY` 設定為 `yes`。

不過，在負責的機構指定正式領域前，請不要設定。即使您有自己的領域而且是由提供者管理的，最好也不要使用，否則BIND 不會轉寄此領域的要求。例如，此領域將無法存取提供者的網頁伺服器。

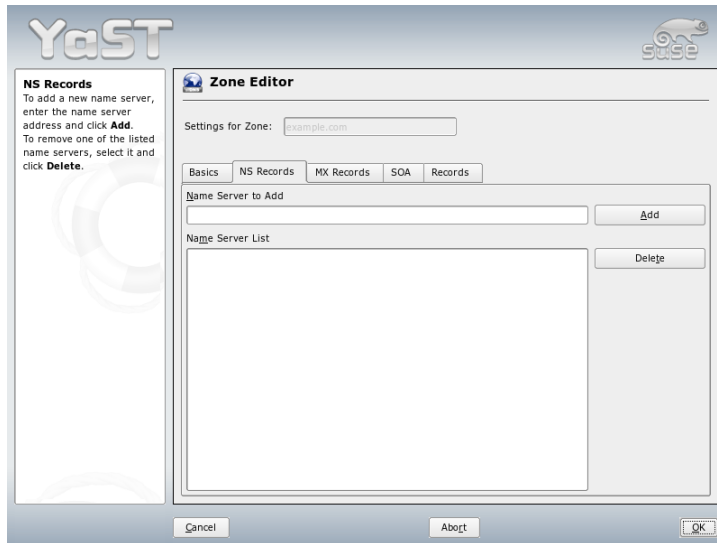


Figure 24.7: DNS 伺服器：區域編輯器(NS 記錄)

若要啓動名稱伺服器，以root 的身份輸入指令`rcnamed start`。如果右邊出現綠色的「完成」，即表示已成功啓動稱爲named 的名稱伺服器程序。使用`host` 或`dig` 程式立即測試本地系統上的名稱伺服器，應該會傳回localhost 作爲預設伺服器，位址爲127.0.0.1。如果不是這種情形，`/etc/resolv.conf` 可能包含不正確的名稱伺服器項目，或是檔案根本不存在。對於第一次測試，輸入`host 127.0.0.1`，這通常都能成功。如果看到錯誤訊息，使用`rcnamed status`，看看伺服器是否真的執行。如果名稱伺服器沒有啓動或是未以預期的方式運作，通常在`/var/log/messages` 記錄檔中可以找到原因。

若要使用提供者的名稱伺服器或將網路上已經執行的名稱伺服器作爲forwarder，請在forwarders 下的options 區段中輸入對應的IP 位址。範例24.1, "named.conf 中的轉寄選項" 中包含的位址只是範例。根據您自己的設定調整這些項目。

Example 24.1: named.conf 中的轉寄選項

```
options {  
    directory "/var/lib/named";
```

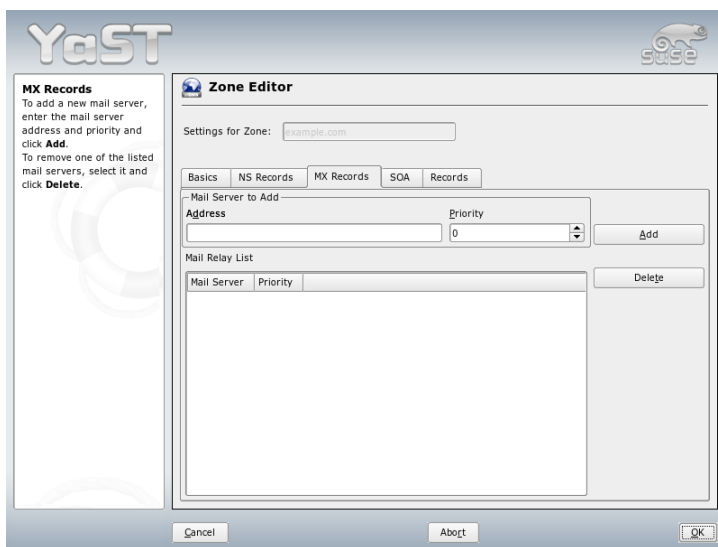


Figure 24.8: DNS 伺服器：區域編輯器(MX 記錄)

```
forwarders { 10.11.12.13; 10.11.12.14; };  
listen-on { 127.0.0.1; 192.168.0.99; };  
allow-query { 127/8; 192.168.0/24; };  
notify no;  
};
```

options 項目後面跟著是區域的項目、localhost 以及 0.0.127.in-addr.arpa。在“.” 之下的 type hint 項目應該永遠是存在的。不需要修改對應檔案，而且應該依其原狀運作。另外也請確定每個項目的前後都有「;」，且大括號在正確的位置。變更組態檔/etc/named.conf 或區域檔後，使用 `rndc reload` 告訴 BIND 重新讀取這些檔案。使用 `rndc restart` 停止和重新啓動名稱伺服器會達成相同的結果。輸入 `rndc stop` 隨時停止伺服器。

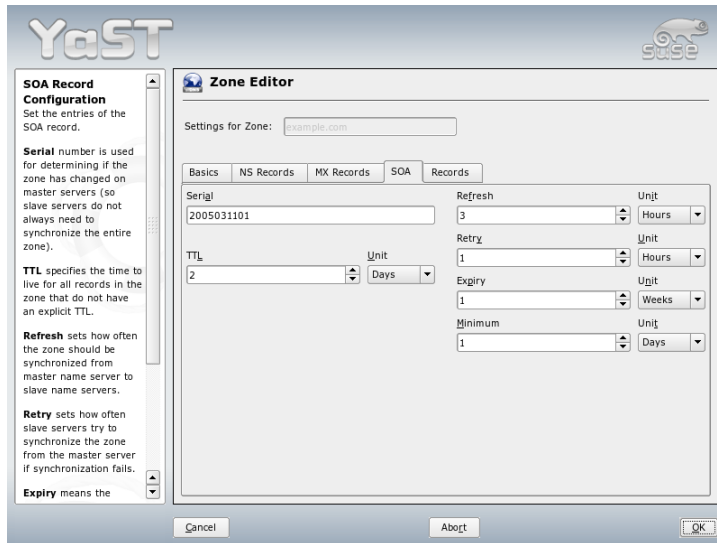


Figure 24.9: DNS 伺服器：區域編輯器(SOA)

24.3 組態檔/etc/named.conf

BIND 名稱伺服器本身的所有設定儲存於檔案 `/etc/named.conf`。不過，要處理之領域的區域資料，包括主機名稱、IP 位址等，儲存於 `/var/lib/named` 目錄中的不同檔案中。詳細資訊會在下面進一步說明。

`/etc/named.conf` 粗略分為兩個部分。其中一個是一般設定的 `options` 區段，另一個則是由個別領域的 `zone` 項目組成。`logging` 區段和 `acl` (存取控制清單) 項目是選擇性的。註解行的開頭是 `#` 符號或 `//`。在範例 24.2, "基本的 `/etc/named.conf`" 中顯示了最基本的 `/etc/named.conf`。

Example 24.2: 基本的 `/etc/named.conf`

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};
```



```
zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};
```

24.3.1 重要組態選項

"*<filename>*" 目錄; 指定BIND 可以在其中找尋包含區域資料檔案的目錄。通常是 `/var/lib/named`。

forwarders { *<ip-address>*; }; 指定無法直接解析DNS 要求時應該轉寄的目標名稱伺服器(大部分是提供者)。以 `10.0.0.1` 之類的IP 位址取代 *<ip-address>*。

forward first; 將會在嘗試透過根名稱伺服器解析DNS 要求前先加以轉寄。除了 `forward first`，可以寫入 `forward only` 轉寄所有要求，而不會有任何要求傳送到根名稱伺服器。這對於防火牆組態是可以理解的。

listen-on port 53 { 127.0.0.1; *<ip-address>*; };
告訴BIND 哪個網路介面和哪個連接埠要接受用戶端查詢。port 53 不需要明確指定，因為53 是預設連接埠。輸入 `127.0.0.1` 允許來自本地主機的要求。如果完全省略此項目，預設會使用所有介面。

listen-on-v6 port 53 { any; }; 告訴BIND 哪個連接埠應該聆聽IPv6 用戶端要求。除了 `any` 外只能使用 `none`。就IPv6 而言，伺服器僅接受萬用字元位址。

query-source address * port 53; 如果防火牆封鎖外送DNS 要求，則需要這個項目。這樣會告訴BIND 從外部的連接埠53 張貼要求，而不是從任何高於1024 的連接埠張貼。

query-source-v6 address * port 53; 告訴BIND 哪個連接埠用於IPv6 查詢。

allow-query { 127.0.0.1; <net>;; 定義用戶端可以張貼DNS要求的網路。使用如192.168.1/24的位址資訊取代<net>。尾端的/24是網路遮罩的縮寫表示式，在此例中為255.255.255.0。

allow-transfer !*;; 控制哪個主機可以要求區域傳輸。在範例中，這類要求是使用!*完全拒絕。沒有這個項目，可以從任一處要求區域傳輸，沒有限制。

statistics-interval 0; 如果沒有這個項目，BIND每小時在/var/log/messages中會產生數行統計資訊。指定0則完全不會顯示這些統計數字，或設定以分鐘為單位的間隔時間。

cleaning-interval 720; 這個選項定義BIND清除其快取的時間間隔。這樣每次清除時會觸發/var/log/messages中的項目。時間單位為分鐘。預設值是60分鐘。

interface-interval 0; BIND會定期搜尋網路介面，尋找新的或不存在的介面。如果此值設定為0，就不會執行這個動作，且BIND僅會聆聽啓動時偵測到的介面。此外，請以分鐘為單位定義間隔時間。預設值是60分鐘。

notify no; 當變更區域資料或重新啓動名稱伺服器時，no會防止通知其他名稱伺服器。

24.3.2 記錄

記錄的內容、方式及位置可以廣泛地在BIND中設定。一般而言，預設設定應該足夠。範例24.3, “關閉記錄的項目”顯示出這類項目的最簡單格式，而且完全停用任何記錄。

Example 24.3: 關閉記錄的項目

```
logging {  
    category default { null; };  
};
```

24.3.3 區域項目

Example 24.4: my-domain.de 的區域項目

```
zone "my-domain.de" in {
    type master;
    file "my-domain.zone";
    notify no;
};
```

在zone之後，指定要管理的領域名稱(my-domain.de)，後面跟著in以及大括號包住的相關選項區塊，如範例24.4, "my-domain.de 的區域項目" 中所示。若要定義slave zone，切換type為slave並指定管理此區域的名稱伺服器為master (也可能成為另一個主要的從屬)，如範例24.5, "other-domain.de 的區域項目" 中所示。

Example 24.5: other-domain.de 的區域項目

```
zone "other-domain.de" in {
    type slave;
    file "slave/other-domain.zone";
    masters { 10.0.0.1; };
};
```

區域選項：

type master; 藉由指定master，告訴BIND區域由本地名稱伺服器處理。這假設區域檔案已經以正確格式建立。

type slave; 此區域傳輸自另一個名稱伺服器。必須與masters一起使用。

type hint; hint類型的區域。用來設定根名稱伺服器。此區域定義可以維持原狀。

file my-domain.zone or file "slave/other-domain.zone";

此項目指定領域之區域資料所在的檔案。從屬並不需要這個檔案，因為可從另一個名稱伺服器取得此資料。若要分別主要和從屬檔案，請為從屬檔案使用目錄slave。

masters { <server-ip-address>;}; 僅從屬區域需要此項目。它指定應該傳輸區域檔案的名稱伺服器。

allow-update {!*}; 此選項控制外部寫入存取，將允許用戶端產生DNS項目，通常由於安全性的緣故不需要此項目。沒有此項目，則完全不允許區域更新。以下項目會產生相同的結果，因為!*有效地禁止任何這類活動。

24.4 區域檔案

需要兩種類型的區域檔案。其中一個指定IP位址給主機名稱，而另一個則相反：提供主機名稱給IP位址。

在區域檔案中使用點符號

。在區域檔案中具有重要意義。如果主機名稱最後不用點符號結尾，會附加區域。指定完整領域名稱的主機名稱必須以點符號結尾，才能避免再次附加領域。缺少點符號或錯置其位置最經常造成名稱伺服器組態錯誤。

第一個要考慮的情況是負責領域world.cosmos的區域檔案world.zone，如範例24.6，“檔案/var/lib/named/world.zone”中所示。

Example 24.6: 檔案/var/lib/named/world.zone

```
1 $TTL 2D
2 world.cosmos. IN SOA      gateway root.world.cosmos. (
3                       2003072441 ; serial
4                       1D         ; refresh
5                       2H         ; retry
6                       1W         ; expiry
7                       2D )       ; minimum
8
9                       IN NS      gateway
10                      IN MX      10 sun
11
12 gateway              IN A        192.168.0.1
13                      IN A        192.168.1.1
14 sun                  IN A        192.168.0.2
15 moon                 IN A        192.168.0.3
16 earth                IN A        192.168.1.2
17 mars                 IN A        192.168.1.3
18 www                  IN CNAME     moon
```

行1：\$TTL 定義應該套用到此檔案中所有項目的預設有效時間。在此範例中，項目的有效時間是兩天(2 D)。

行2： 這是SOA (授權開始) 控制記錄開始的地方：

- 要管理的領域名稱是第一個位置的`world.cosmos`。這以`.`結尾，否則會第二次附加區域。或者，可以在此輸入`@`，這樣會從`/etc/named.conf` 中的對應項目擷取區域。
- 在`IN SOA` 之後是名稱伺服器的名稱，做為此區域的主要伺服器。名稱會從`gateway` 擴展為`gateway.world.cosmos`，因為它沒有用`.`結尾。
- 後面跟著負責此名稱伺服器之人員的電子郵件地址。因為`@`符號已經具有特殊意義，所以在此輸入`.`來代替。對於`root@world.cosmos`，項目必須讀成`root.world.cosmos.`。必須包含在結尾，防止新增區域。
- (將所有到) 的行都包含在SOA 記錄中。

行3： `serial number` 是任意號碼，每次變更此檔案時就會增加。需要通知次要名稱伺服器(從屬伺服器) 這種變更。對於這種情形，十個數字的日期及執行號碼，寫法是`YYYYMMDDNN`，已成為習慣格式。

行4： `refresh rate` 指定次要名稱伺服器確認區域`serial number` 的時間間隔。在此例中，是一天。

行5： `retry rate` 指定次要名稱伺服器在發生錯誤時嘗試再次連絡主要伺服器的時間間隔。在此例中，是兩小時。

行6： `expiration time` 指定次要名稱伺服器沒有重新取得與主要伺服器的連絡時捨棄快取資料後的時間範圍。在此例中，是一週。

行7： SOA 記錄中的最後一個項目指定`negative caching TTL`，亦即其他伺服器未解析DNS 查詢之結果可以快取的時間。

行9： `IN NS` 指定負責此領域的名稱伺服器。`gateway` 會擴展為`gateway.world.cosmos`，因為它沒有以`.`結尾。可能有數行會像這樣，其中一行是主要名稱伺服器，而每個次要名稱伺服器各一行。如果`/etc/named.conf` 中的`notify` 不是設定為`no`，此處列出的所有名稱伺服器會收到區域資料變更的通知。

行10： `MX` 記錄指定接收、處理和轉寄領域`world.cosmos` 電子郵件的郵件伺服器。在此範例中，是主機`sun.world.cosmos`。主機名稱前的號碼是優先設定值。如果有多個`MX` 項目，會先優先使用具有最小值的郵件伺服器，而如果郵件無法送到此伺服器，就會嘗試使用下一個較高的值。

行12-17：這些是指定給主機名稱的一或多個IP位址的實際位址記錄。列示於此的名稱沒有.，因為沒有包含其領域，所以會在後面加上world.cosmos。將兩個IP位址指定給主機gateway，因為它有兩張網路卡。如果主機位址是傳統位址(IPv4)，記錄會使用A標示。如果位址是IPv6位址，項目會使用A6標示。IPv6位址之前的記錄是AAAA，現在已廢除不用。

行18：別名www可以用來定址mond (CNAME表示*canonical name* (標準名稱))。

擬領域in-addr.arpa用來反向查詢IP位址到主機名稱。它會以反向標記法附加到位址的網路部分。因此192.168.1會解析為1.168.192.in-addr.arpa。請參閱範例24.7, "反向查詢"。

Example 24.7: 反向查詢

```
1
2 $TTL 2D
3 1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
4                               2003072441           ; serial
5                               1D                     ; refresh
6                               2H                     ; retry
7                               1W                     ; expiry
8                               2D )                   ; minimum
9
10                              IN NS                 gateway.world.cosmos.
11
12 1                              IN PTR              gateway.world.cosmos.
13 2                              IN PTR              earth.world.cosmos.
14 3                              IN PTR              mars.world.cosmos.
```

行1：\$TTL定義套用到此處所有項目的標準TTL。

行2：組態檔應該為網路192.168.1.0啓用反向查詢。假設區域稱為1.168.192.in-addr.arpa，則不應該新增到主機名稱。因此，以完整格式輸入所有主機名稱，亦即包含領域以及結尾的.。其餘的項目與之前world.cosmos範例中所述的項目相同。

行3-7：請參閱之前的world.cosmos範例。

行9：同樣地，此行指定負責此區域的名稱伺服器。不過，這一次，以完整格式輸入名稱，亦即包含領域以及結尾的.。

行11-13：這些是相關主機上IP位址的指標記錄提示。行的開頭僅會輸入IP位址的最後一個部分，結尾不加上.。對此附加區域(不加上.in-addr.arpa)會造成完整IP位址變成反向順序。

通常，不同BIND版本之間的區域傳輸應該是没有問題的。

24.5 區域資料的動態更新

「動態更新」(*dynamic update*) 這個詞是指新增、變更或刪除主要伺服器的區域檔案中項目的作業。此機制於RFC 2136 中有詳細描述。利用新增選擇性的allow-update 或update-policy 規則，來為每個區域項目個別設定動態更新。動態更新的區域不應該手動修改。

使用指令nsupdate 將要更新的項目傳送到伺服器。對於此指令的完整語法，請查閱nsupdate 的手冊頁(man 8 nsupdate)。為了安全性的緣故，這類更新應該使用TSIG 金鑰加以執行，如節24.6, “安全交易” 所述。

24.6 安全交易

安全交易可以透過採用共用秘密金鑰(也稱為TSIG 金鑰) 的交易簽名(TSIG) 來完成。本節說明如何產生及使用這類金鑰。

對於不同伺服器之間的通訊以及動態更新區域資料而言，是需要安全交易的。讓存取控制依靠金鑰比單純依靠IP 位址要來得安全許多。

使用以下指令產生TSIG 金鑰(有關詳細資訊，請參閱man dnssec-keygen)：

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

這樣會建立兩個檔案，名稱類似如下：

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

金鑰本身(如ejIkuCyyGJwwuN3xAteKgg== 的字串) 在兩個檔案中都可找到。如果要用於交易，第二個檔案(Khost1-host2.+157+34265.key) 必須傳輸到遠端主機，最好是以安全的方式傳輸(例如，使用scp)。在遠端伺服器上，金鑰必須包含於檔案/etc/named.conf 內，才能開啓host1 與host2 之間的安全通訊：

```
key host1-host2. {  
    algorithm hmac-md5;  
    secret "ejIkuCyyGJwwuN3xAteKgg==";  
};
```



`/etc/named.conf` 的檔案權限

確定`/etc/named.conf` 的權限受到適當的限制。此檔案的預設值是0640，擁有者為root及群組named。另一種方法是，將金鑰移到具有特殊限定權限的額外檔案，是後續從`/etc/named.conf` 包含進來的。



若要讓伺服器host1 能夠使用host2 (在此範例中位址為192.168.2.3) 的金鑰，伺服器的`/etc/named.conf` 必須包含以下規則：

```
server 192.168.2.3 {  
    keys { host1-host2. ;};  
};
```

類比項目必須包含於host2 的組態檔中。

針對IP 位址和位址範圍定義的任何ACL (存取控制清單，不得與檔案系統ACL 混淆) 來新增TSIG 金鑰，以啓用異動保全性。對應項目應該看起來如下：

```
allow-update { key host1-host2. ;};
```

此主題在update-policy 下的*BIND Administrator Reference Manual* 中有詳細討論。

24.7 DNS 安全性

DNSSEC 或DNS 安全性細述於RFC 2535。DNSSEC 的可用工具在BIND 手冊中有詳加討論。

視為安全的區域必須具有關聯的一或多個區域金鑰。這些金鑰是使用`dnssec-keygen` 產生，如同主機金鑰一樣。目前是使用DSA 加密演算法產生這些金鑰。產生的公用金鑰應該包含於套用`$INCLUDE` 規則的對應區域檔案中。

藉由指令`dnssec-makekeyset`，產生的所有金鑰會封裝為一組，然後必須透過安全的方法傳輸到父區域。在父區域上，會使用`dnssec-signkey` 簽署該組金鑰。接著會使用透過此指令產生的檔案，以`dnssec-signzone` 指令簽署區域，然後會為`/etc/named.conf` 中每個區域產生要包含在內的檔案。

24.8 若需詳細資訊

如需其他資訊，請參閱 *BIND Administrator Reference Manual*，安裝於 `/usr/share/doc/packages/bind/` 下。另外也請參閱手冊參考的RFC 以及 BIND 隨附的手冊頁。有關 SUSE LINUX 中 BIND 的最新資訊，請參閱 `/usr/share/doc/packages/bind/README.SuSE`。

使 用NIS

一旦網路中的多個UNIX系統想要存取常用資源，所有使用者和群組身分對於該網路中所有機器均相同就變得很重要。網路應該對使用者透明化：無論使用者是使用哪些機器，他們永遠都可以在相同環境中找到自己。這可藉由NIS和NFS服務來達成。NFS透過網路配送檔案系統，將在章26, 使用NFS共用檔案系統中討論。

NIS(網路資訊服務)可描述成像資料庫的服務，提供透過網路對/etc/passwd、/etc/shadow和/etc/group內容的存取權。NIS也可用於其他用途(例如，提供如/etc/hosts或/etc/services檔案的內容以供使用)，但這超過本介紹的範圍。人們通常將NIS視為YP，因它與網路「黃頁」的概念相似。

25.1	設定NIS 伺服器	386
25.2	設定NIS 用戶端	388

25.1 設定NIS 伺服器

若要組態，從YaST 模組‘網路服務’ 選取‘NIS 伺服器’。如果目前在您的網路中沒有NIS 伺服器存在，請在下一個畫面中啓用‘安裝與設定主要NIS 伺服器’。如果您已經有NIS 伺服器(主要伺服器)，便可以新增NIS 次要伺服器(例如，如果您想要設定新的子網路)。首先，主要伺服器的組態如描述。

如果遺失部分需要的套件，請放入要求的CD 或DVD 以自動安裝套件。在組態對話方塊的頂端輸入領域名稱，如圖形25.1, “NIS 伺服器組態工具” 中所示。以核取方塊定義主機是否也是NIS 用戶端，可讓使用者登入並從NIS 伺服器存取資料。

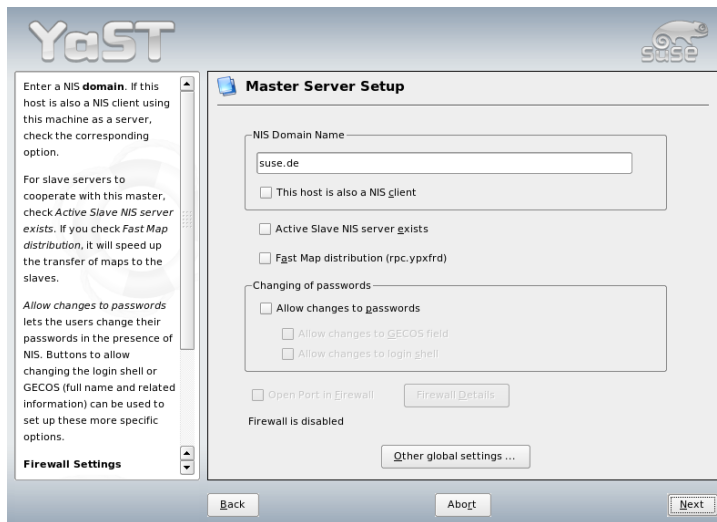


Figure 25.1: NIS 伺服器組態工具

若要在之後設定網路中其他NIS 伺服器(次要伺服器)，請現在啓用‘啓用現有的次要NIS 伺服器’。選取‘快速對應配送’ 以設定從主要到次要伺服器的快速資料庫項目傳輸。

若要讓您的網路中的使用者(包括本地使用者和透過NIS 伺服器管理的使用者) 變更有NIS 伺服器的密碼(以指令 `yppasswd`)，請啓用對應選項。如此便可使用‘允許變更GECOS 欄位’ 和‘允許變更登入外圍程序’。“GECOS” 表示使用者可使用指

令ypchfn 變更其名稱與位址設定。例如，“SHELL”讓使用者以ypchsh 指令變更其預設外圍程序，以便從bash 切換到sh。

按一下‘其他全域設定’存取如圖形25.2, “變更NIS 伺服器的目錄與同步化檔案” 中所示的畫面，可在其中變更NIS 伺服器的來源目錄(預設為/etc)。此外，在此處可合併密碼與群組。設定值應該為‘是’，如此檔案(/etc/passwd、/etc/shadow 和 /etc/group) 便可同步。同時需決定最小使用者與群組ID。按‘確定’以確認您的設定值並回到上一個畫面。然後按‘下一步’。

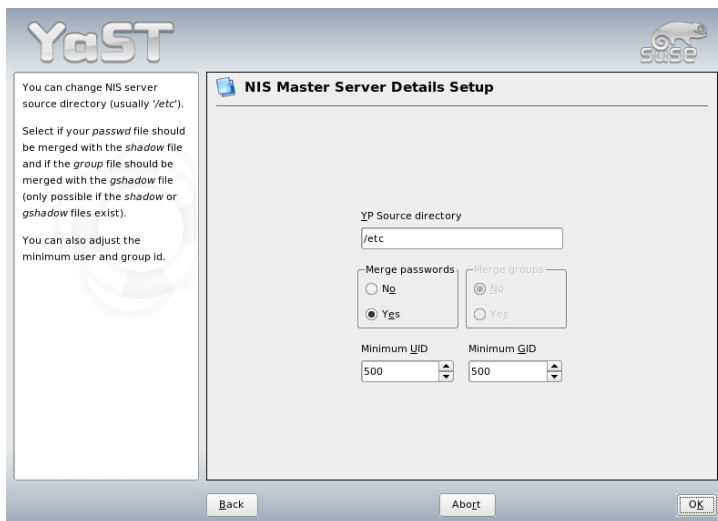


Figure 25.2: 變更NIS 伺服器的目錄與同步化檔案

如果您先前啓用了‘可用的從屬NIS 伺服器存在’，請輸入要做為次要伺服器的主機名稱然後按‘下一步’。如果您不使用次要伺服器，會略過次要伺服器組態，直接繼續資料庫組態的對話方塊。此處，請指定映射，從NIS 伺服器傳輸到用戶端的部分資料庫。預設值通常足夠使用。

‘下一步’會繼續最後一個對話方塊，如圖形25.3, “設定NIS 伺服器的要求權限” 中所示。指定可從哪些網路傳送要求至NIS 伺服器。通常，這是您的內部網路。在此例中，應該是下列兩個項目：

```
255.0.0.0      127.0.0.0 0.0.0.0      0.0.0.0
```

第一個讓您可從自己的主機連線，即NIS 伺服器。第二個讓所有具有相同網路存取權的主機，可以傳送要求至伺服器。

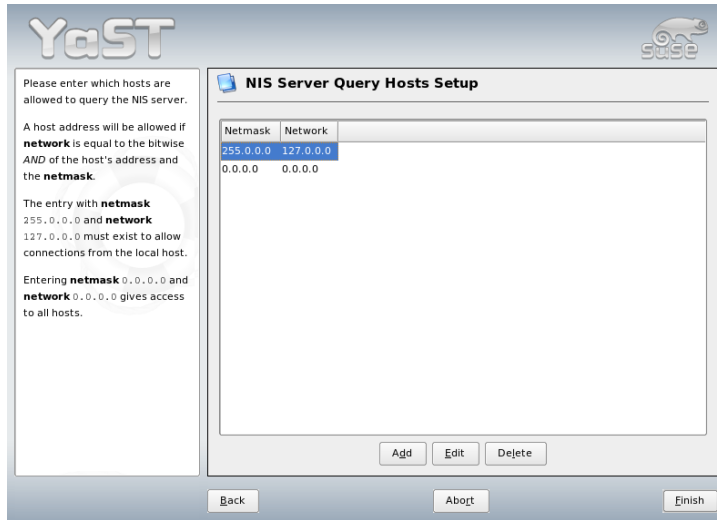


Figure 25.3: 設定NIS 伺服器的要求權限

自動防火牆組態

如果您的系統上有防火牆在作用中(SuSEfirewall2)，YaST 會在選取‘在防火牆中開啓埠’時，啓用portmap 服務以調整其NIS 伺服器的組態。

25.2 設定NIS 用戶端

使用此模組以設定NIS 用戶端。在選擇使用NIS 以及自動掛載器(視狀況而定) 之後，會開啓此對話方塊。選擇主機是否有固定IP 位址，或是要接受DHCP 所發出的IP 位址。DHCP 也提供NIS 領域和NIS 伺服器。若需有關DHCP 的詳細資訊，請參閱章27, DHCP。如果使用靜態IP 位址，請手動指定NIS 領域和NIS 伺服器。請參閱圖形25.4, “設定NIS 伺服器的領域和位址”。‘尋找’可讓YaST 搜尋網路中作用中的NIS 伺服器。

此外，您可以使用一個預設領域指定多個領域。使用‘新增’以指定包括個別領域廣播功能的多個伺服器。

在進階設定中，如果您不想要其他主機查詢您的用戶端所使用的伺服器，請勾選‘僅回答本地主機’。勾選‘中斷伺服器’，則用戶端可接收透過未授權的連接埠通訊的伺服器之回覆。若需進一步資訊，請參閱man ypbind。

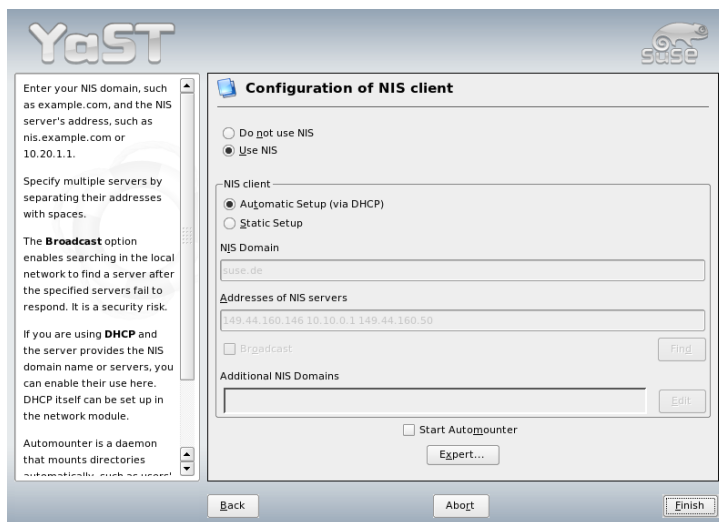


Figure 25.4: 設定NIS 伺服器的領域和位址

使用NFS 共用檔案系統

如章25, 使用NIS 中所述，與NIS 一起使用的NFS 讓網路對使用者而言透明化。使用NFS，可以透過網路配送檔案系統。使用者登入哪個終端機都沒關係。他永遠可在相同環境中找到自己。

和NIS 相似，NFS 是非對稱服務。有NFS 伺服器和NFS 用戶端。一台機器可同時扮演這兩種角色— 它可透過網路提供檔案系統(輸出)，也可以從其他主機裝載檔案系統(輸入)。一般而言，這些伺服器必須具有非常大的硬碟容量，其他用戶端裝載其檔案系統。

26.1	以YaST 輸入檔案系統	392
26.2	手動輸入檔案系統	392
26.3	以YaST 輸出檔案系統	393
26.4	手動輸出檔案系統	394

DNS 所需

原則上，可以只使用IP 位址來進行所有的輸出。不過，若要避免逾時，您應該具有運作中的DNS 系統。這是基於記錄目的所必備的，因為裝載的精靈會進行反向查詢。

26.1 以YaST 輸入檔案系統

被授權的使用者可從NFS 伺服器裝載NFS 目錄到自己的檔案目錄樹。使用YaST 模組‘NFS 用戶端’是最簡單的達成方式。僅需鍵入NFS 伺服器的主機名稱、要輸入的目錄和在本地裝載此目錄的裝載點。在第一個對話方塊中，按一下‘新增’之後便可完成。請參閱圖形26.1, “使用YaST 的NFS 用戶端組態”。

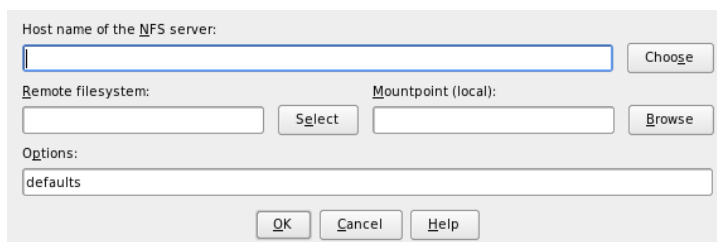


Figure 26.1: 使用YaST 的NFS 用戶端組態

26.2 手動輸入檔案系統

您可輕易地從NFS 伺服器手動輸入檔案系統。唯一的先決條件是執行中的RPC 埠對應程式，以root 使用者身份輸入指令`rpcportmap start` 便可啓動。只要符合此先決條件，便可像在本地硬碟一樣，以下列語法使用指令`mount`，將對應機器上輸出的遠端檔案系統裝載在檔案系統中：

```
mount host:remote-path local-path
```

例如，如果要輸入來自機器sun 的使用者目錄，請使用以下指令：

```
mount sun:/home /home
```

26.3 以YaST 輸出檔案系統

使用YaST，將您網路中的主機轉變為NFS 伺服器—可將目錄和檔案輸出到所有授與存取權的主機之伺服器。提供應用程式給團隊中的所有工作夥伴，但不在每一個主機上本地安裝，可達成此目的。若要安裝此種伺服器，請啟動YaST 並選擇‘網路服務’→‘NFS 伺服器’。會開啓像圖形26.2, “NFS 伺服器組態工具” 中的對話方塊。

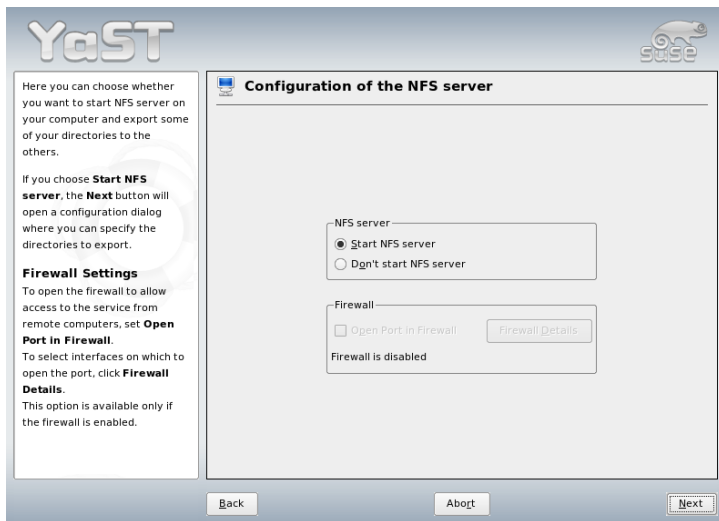


Figure 26.2: NFS 伺服器組態工具

下一步，啓用‘啓動NFS 伺服器’，並按一下‘下一步’。在上方的文字欄位中，鍵入要輸入的目錄。在下方，輸入可以存取它們的主機。在圖形26.3, “以YaST 設定NFS 伺服器” 中顯示了此對話方塊。每一台主機有四個選項可設定：單一主機、網路群組、萬用字元和IP 網路。man exports 提供這些選項更詳盡的說明。‘結束’ 可完成組態。

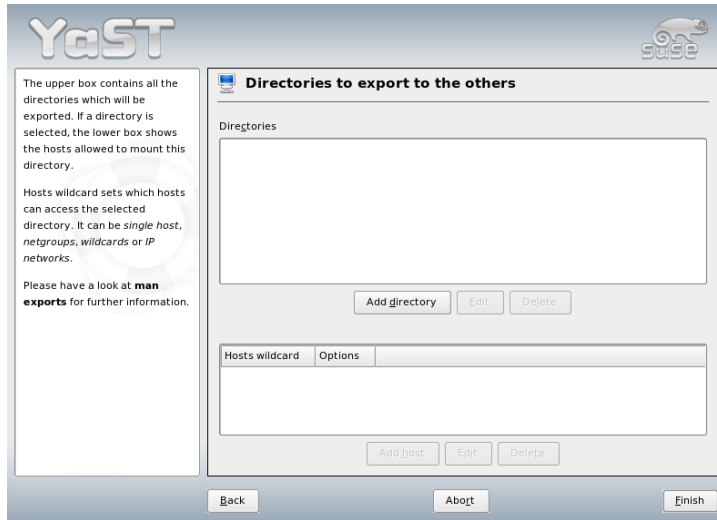


Figure 26.3: 以YaST 設定NFS 伺服器

自動防火牆組態

如果您的系統上有防火牆在作用中(SuSEfirewall2)，YaST 會在選取‘在防火牆中開啓埠’時，啓用nfs 服務以調整其NFS 伺服器組態。

26.4 手動輸出檔案系統

如果您不想使用YaST，請確定NFS 伺服器上有執行下列系統：

- RPC portmapper (portmap)
- RPC 裝載精靈(rpc.mountd)
- RPC NFS 精靈(rpc.nfsd)

若要在系統啓動時使用程序檔 `/etc/init.d/portmap` 和 `/etc/init.d/nfsserver` 來啓動這些服務，請輸入指令 `insserv /etc/init.d/nfsserver` 和 `insserv /etc/init.d/portmap`。同時在組態檔 `/etc/exports` 中定義哪些檔案系統應輸出至哪些主機。

對每一個要輸出的目錄，必須有一行來設定哪些主機可以何種權限存取該目錄。此目錄中的所有子目錄也會自動輸出。被授權的機器通常會以完整名稱來指定(包括領域名稱)，但是也可能會使用萬用字元，如 `*` 或 `?` (以 Bash 外圍程式相同方式擴展)。如果在此處沒有指定機器，任何機器均可以指定權限輸入此檔案系統。

在機器名稱之後以括號設定輸出檔案系統的權限。最重要的選項顯示於表格 26.1, “輸出檔案系統的權限” 中。

Table 26.1: 輸出檔案系統的權限

選項	意義
<code>ro</code>	檔案系統以唯讀權限輸出(預設值)。
<code>rw</code>	檔案系統以讀寫權限輸出。
<code>root_squash</code>	這將可確保輸入機器的 <code>root</code> 使用者在此檔案系統上沒有 <code>root</code> 權限。可藉由將使用者 ID 65534 指定給具有使用者 ID 0 (<code>root</code>) 的使用者來達成。此使用者 ID 應設為 <code>nobody</code> (預設值)。
<code>no_root_squash</code>	請勿將使用者 ID 0 指定為使用者 ID 65534，請維持 <code>root</code> 權限的有效性。
<code>link_relative</code>	將絕對連結(這些是以 <code>/</code> 開頭)轉換為 <code>../</code> 的順序。只有在裝載機器的整個檔案系統時才有用(預設值)。
<code>link_absolute</code>	符號連結保留不動。
<code>map_identity</code>	使用者 ID 在用戶端與伺服器上是完全相同的(預設值)。
<code>map_daemon</code>	用戶端與伺服器沒有相符的使用者 ID。這會讓 <code>nfdsd</code> 建立使用者 ID 的轉換表。此工作必須有 <code>ugidd</code> 精靈。

您的 `exports` 檔案看起來會像範例 26.1, “`/etc/exports`”。`/etc/exports` 是由 `mountd` 與 `nfdsd` 讀取。如果您對此檔案做任何變更，請重新啓動 `mountd` 和 `nfdsd` 讓變更生效。使用 `rcnfsserver restart` 可輕易完成。

Example 26.1: /etc/exports

```
#  
# /etc/exports  
#  
/home          sun(rw)   venus(rw)  
/usr/X11       sun(ro)   venus(ro)  
/usr/lib/texmf sun(ro)   venus(rw)  
/              earth(ro,root_squash)  
/home/ftp      (ro)  
# End of exports
```

DHCP

動態主機組態通訊協定 (DHCP, dynamic host configuration protocol) 的用途是爲了從伺服器集中指定網路設定，而不是在每個工作站上分別設定。設定要使用DHCP的主機對於自己的靜態位址並沒有控制權。它可以根據伺服器的指示完全且自動地設定自己本身。

27.1	使用YaST 設定DHCP 伺服器	398
27.2	DHCP 軟體套件	399
27.3	DHCP 伺服器dhcpd	400
27.4	若需詳細資訊	404

使用DHCP的一個方式是識別每個使用網路卡硬體位址(大部份的情況下是固定的)的用戶端，以後每次用戶端連接伺服器時，將相同的設定值提供給用戶端。或者DHCP也可以設定成讓伺服器從專門使用的位址集區動態指定位址給每個有關的用戶端。在後者的情形中，DHCP伺服器在每次從用戶端接受到要求時指定相同的位址給用戶端，即使經過較長的時間也是一樣。當然只有在網路的用戶端數量不超過位址數量時才能這樣運作。

由於這些可以採用的方法，因此DHCP透過這兩個方式讓系統管理員的工作變得更容易。任何有關位址及一般網路組態的變更(甚至更大的變更)，只要編輯伺服器的組態檔，即可集中執行。這樣比重新設定許多工作站便利許多。另外也更易於整合機器(特別是新機器)到網路中，因為可以從集區對其指定IP位址。從DHCP伺服器擷取適當網路設定，對於經常使用不同網路的筆記型電腦特別有用。

DHCP伺服器不僅提供IP位址和網路遮罩，也提供主機名稱、領域名稱、閘道和名稱伺服器位址供用戶端使用。除此之外，DHCP也允許集中設定一些其他的參數，例如，用戶端可以輪詢目前時間的時間伺服器或甚至是列印伺服器。

27.1 使用YaST 設定DHCP 伺服器

第一次啓動模組後，YaST 會啓動四步驟的組態精靈。完成此精靈可以設定基本DHCP伺服器。

選取網路介面 在第一步驟中，YaST 會尋找系統上可用的網路介面，然後顯示清單。從清單中，選取DHCP伺服器要傾聽的介面，並選取‘開啓選取介面的防火牆’以開啓此介面的防火牆。請參閱圖形27.1, “DHCP伺服器：選取網路介面”。

全域設定 在輸入欄位，提供DHCP伺服器應該管理的所有用戶端的網路細節。這些細節包括領域名稱、時間伺服器的位址、主要及次要名稱伺服器的位址、列印和WINS伺服器的位址(供同時具有Windows與Linux用戶端的混和網路使用)、閘道位址以及租用時間。請參閱圖形27.2, “DHCP伺服器：全域設定”。

動態DHCP 在此步驟中，設定應該如何將動態IP位址指定給用戶端。若要這樣做，指定伺服器可以指定位址給DHCP用戶端的IP範圍。所有這些位址應該涵蓋在相同的網路遮罩下。另外也指定用戶端可以保留其IP位址、不用要求延續出租的時間限制。或者，指定最長出租時間，也就是伺服器保留特定用戶端之IP位址的時間。請參閱圖形27.3, “DHCP伺服器：動態DHCP”。

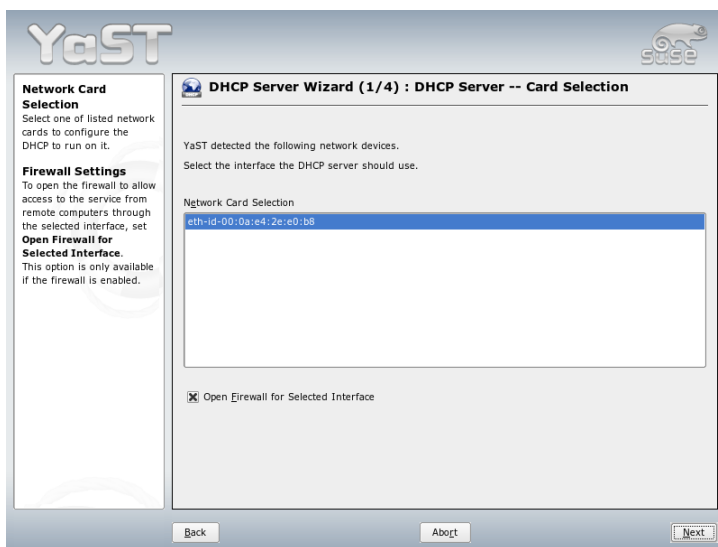


Figure 27.1: DHCP 伺服器：選取網路介面

完成組態及設定啓動模式。在組態精靈的第三步驟後，會顯示最後一個對話方塊，定義DHCP 伺服器的啓動方式。在此您可以決定是否在系統開機時自動啓動DHCP 伺服器，或在需要時才手動啓動(例如，爲了測試)。按一下‘完成’完成伺服器的組態。請參閱圖形27.4, “DHCP 伺服器：啓動”。

27.2 DHCP 軟體套件

DHCP 伺服器與DHCP 用戶端皆可用於SUSE LINUX。可用的DHCP 伺服器是dhcpcd (由Internet Software Consortium 發佈)。在用戶端，選擇兩種不同的DHCP 用戶端程式：dhclient (也是ISC 發佈的) 以及dhcpcd 套件中的DHCP 用戶端精靈。

SUSE LINUX 預設會安裝dhcpcd。該程式易於使用，而且每次系統開機時就會自動啓動來等待DHCP 伺服器。它不需要組態檔來執行其工作，而且大部份的標準設定可以直接使用。如果是更複雜的情形，請使用ISC dhclient，這是透過組態檔/etc/dhclient.conf 來控制的。

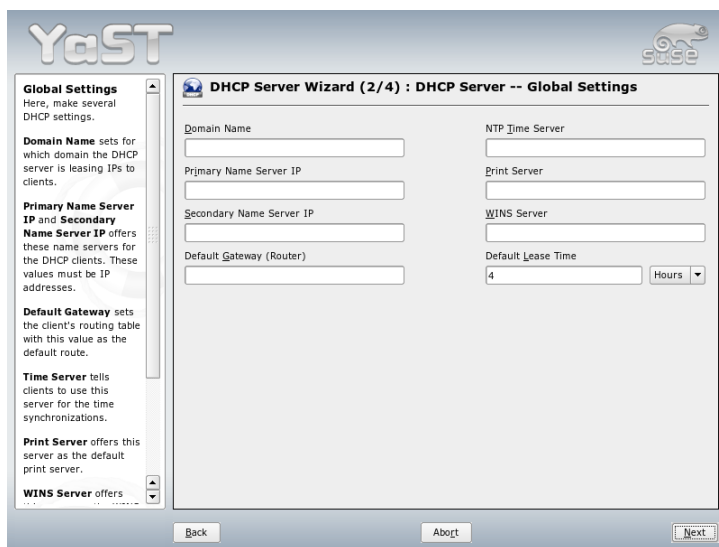


Figure 27.2: DHCP 伺服器：全域設定

27.3 DHCP 伺服器dhcpd

任何DHCP系統的核心都是動態主機組態通訊協定精靈。這個伺服器會依照組態檔/etc/dhcpd.conf中定義的設定，出租位址並監看其使用情形。藉由變更此檔案中的參數及值，系統管理員可以透過數種方式影響程式的行為。請參閱範例27.1, “組態檔/etc/dhcpd.conf”中的/etc/dhcpd.conf基本範例檔案。

Example 27.1: 組態檔/etc/dhcpd.conf

```
default-lease-time 600;           # 10 minutes
max-lease-time 7200;             # 2 hours

option domain-name "cosmos.all";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
```

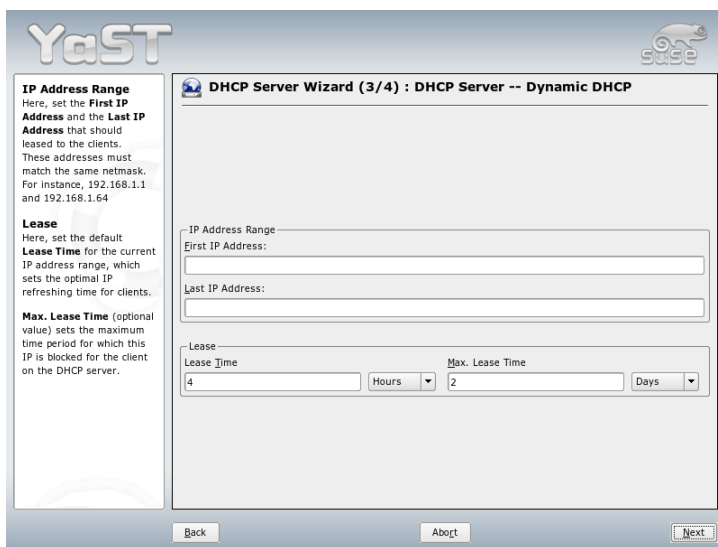


Figure 27.3: DHCP 伺服器：動態DHCP

```
{  
  range 192.168.1.10 192.168.1.20;  
  range 192.168.1.100 192.168.1.200;  
}
```

這個簡單的組態檔應足以讓DHCP 伺服器指定網路中的IP 位址。請確定每行結尾都插入分號，否則將不會啟動dhcpd。

上述範例檔可以分為三個部分。第一個部份定義要求用戶端申請續約前IP 位址預設的租用秒數(default-lease-time)。其中也包含機器保留DHCP 伺服器指定之IP 位址、不用申請續約的最長時間的陳述式(max-lease-time)。

在第二部份中，一些基本網路參數定義於全域層級：

- 行option domain-name 定義了您網路的預設領域。
- 利用option domain-name-servers 項目，最多可指定三個DNS 伺服器值，用來解析IP 位址為主機名稱，反之亦然。理想上來說，設定DHCP 前請先設定您機器上或網路上其他地方的名稱伺服器。該名稱伺服器也可

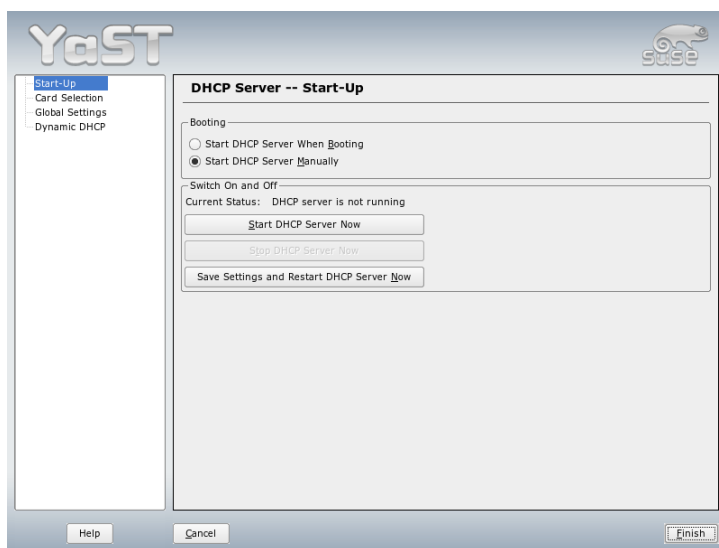


Figure 27.4: DHCP 伺服器：啓動

以為每個動態位址定義主機名稱，反之亦然。若要瞭解如何設定您自己的名稱伺服器，請參閱章24, 領域名稱系統。

- `option broadcast-address` 這一行定義了要求用戶端所使用的廣播位址。
- 利用 `option routers`，告訴伺服器要將無法送至區域網路上主機的資料封包傳送到什麼地方(依據提供的來源和目的主機位址及子網路遮罩)。在大多數的情況中，特別是較小的網路，此路由與網際網路閘道是完全一樣的。
- 利用 `option subnet-mask`，指定指定給用戶端的網路遮罩。

檔案的最後一個部分是為了定義網路，包括子網路遮罩。為了完成，指定DHCP精靈用來指定IP位址給有關用戶端的位址範圍。在此範例中，用戶端的任何指定位址可能介於192.168.1.10與192.168.1.20以及192.168.1.100與192.168.1.200之間。

編輯這幾行後，應該能夠使用指令 `rcdhcpd start` 啓動DHCP精靈。應該是立即就能使用。使用指令 `rcdhcpd check-syntax` 執行簡短的語法檢查。如果在

組態上遇到任何沒有預期的問題，例如伺服器出現錯誤而中止或在啓動時沒有傳回done，則查詢主系統記錄/var/log/messages 或主控台 10 ((Ctrl)-(Alt)-(F10)) 上的資訊，應該即可找出問題所在。

在預設SUSE LINUX 系統上，爲了安全的緣故，會在chroot 環境中啓動DHCP 精靈。組態檔必須複製到chroot 環境，如此精靈才可以找到這些檔案。通常是不需要擔心發生這種情形，因爲指令rcdhcpd start 會自動複製檔案。

27.3.1 使用固定IP 位址的用戶端

如上面所說，DHCP 也可以對每個要求指定預先定義的靜態位址給特定用戶端。明確指定的位址永遠比集區的動態位址優先採用。不僅如此，靜態位址永遠不會過期，與動態位址不同，例如，可用的位址不夠時，伺服器則需要在用戶端間重新散佈位址。

若要識別設定爲「靜態」位址的用戶端，dhcpd 會使用硬體位址，這是全球唯一、固定的數字代碼，由六對八位元組組成，用來識別所有網路設備(例如，00:00:45:12:EE:F4)。如果相對的行(如範例27.2, “組態檔的增加部分” 中所示) 新增到範例27.1, “組態檔/etc/dhcpd.conf” 的組態檔，則DHCP 精靈在任何情況下，永遠會指定相同的資料集給對應的用戶端。

Example 27.2: 組態檔的增加部分

```
host earth {
hardware ethernet 00:00:45:12:EE:F4;
fixed-address 192.168.1.21;
}
```

對應用戶端的名稱(host <hostname>)，此爲earth) 會輸入到第一行，而在第二行則是MAC 位址。在Linux 主機上，使用指令ifstatus 後面接著網路設備(例如，eth0)，可以判定此位址。如有必要，首先使用ifup eth0 啓用網路卡。輸出應該包含如下內容

```
link/ether 00:00:45:12:EE:F4
```

在上述範例中，具有網路卡、且MAC 位址爲00:00:45:12:EE:F4 的用戶端會自動指定IP 位址192.168.1.21 及主機名稱earth。幾乎所有情況中輸入的硬體類型會是ethernet，儘管通常在IBM 系統上找到的是token-ring，也是可以支援的。

27.3.2 SUSE LINUX 版本

爲了改善安全性，ISC 之DHCP 伺服器的SUSE 版本隨附應用了Ari Edelkind 的non-root/chroot 修補程式。這樣可讓dhcpcd 利用使用者ID nobody 執行，而且也能執行於chroot 環境下(/var/lib/dhcp)。如要這樣做，組態檔dhcpcd.conf 必須位於/var/lib/dhcp/etc。init 程序檔啓動時會自動複製檔案到此目錄。

伺服器與此功能相關的行爲，是透過檔案/etc/sysconfig/dhpcpd 中的項目來控制的。若是不要在chroot 環境下執行dhcpcd，請將/etc/sysconfig/dhpcpd 中的DHCPD_RUN_CHROOTED 變數設定爲“no”。

若要讓dhcpcd 從chroot 環境內解析主機名稱，必須也要複製以下其他組態檔案：

- /etc/localtime
- /etc/host.conf
- /etc/hosts
- /etc/resolv.conf

啓動init 程序檔時，這些檔案會複製到/var/lib/dhcp/etc/。如果/etc/ppp/ip-up 之類的程序檔動態修改這些檔案時，則進行所需變更時，也要考慮到這些複本。不過，如果組態檔僅指定IP 位址(而不是指定主機名稱)時，則不需要擔心。

如果您的組態包括應該要複製到chroot 環境的其他檔案，請在檔案etc/sysconfig/dhpcpd 中的變數DHCPD_CONF_INCLUDE_FILES 下指定這些檔案。若是要確定重新啓動系統記錄精靈時DHCP 記錄設備仍可以運作，必須要在檔案/etc/sysconfig/syslog 中變數SYSLOGD_PARAMS 下新增選項“-a /var/lib/dhcp/dev/log”。

27.4 若需詳細資訊

有關DHCP 的詳細資訊請參閱Internet Software Consortium (<http://www.isc.org/products/DHCP/>) 的網站。在dhcpcd、dhcpcd.conf、dhcpcd.leases 以及dhcp-options 中的手冊頁也可以找到資訊。

使用xntp 進行時間同步化

NTP (網路時間協定) 機制是透過網路同步化系統時間的協定。首先，機器可以從提供可靠時間來源的伺服器取得時間。其次，機器本身在網路中可以做為其他電腦的時間來源。這個目標是雙重的，即維護絕對正確的時間，並同步化網路內所有機器的系統時間。

28.1	設定網路中的xntp	406
28.2	設定本機參考時鐘	406
28.3	使用YaST 設定NTP 用戶端	407

維護精準的系統時間對於許多情況都非重要。內建的硬體(BIOS) 時鐘通常無法符合像是資料庫等應用程式的需求。手動校正系統時間有可能會造成嚴重的問題，因為，例如時間倒退將可能造成重要應用程式無法正常運作。在網路中，通常有必要同步化所有機器的系統時間，但手動調整時間並非理想的方法。xntp 提供解決這些問題的機制。它會透過網路中可靠的時間伺服器來持續調整系統時間。它可以進一步管理本機參考的時鐘，例如收音機控制的時鐘。

28.1 設定網路中的xntp

xntp 會預先設定為使用本機電腦時鐘做為時間參考。不過，使用(BIOS) 時鐘，僅供萬一沒有更精確的時間來源時備用。在網路上使用時間伺服器的最簡單方式就是設定伺服器參數。例如，如果網路中有某個名稱為ntp.example.com的可用時間伺服器，請將它的名稱，即server ntp.example.com加入/etc/ntp.conf 檔案。若要新增更多時間伺服器，請以關鍵字伺服器插入其他的行。在以rcxntpd start 指令啓始xntpd 後，將需要一個小時才能使時間穩定下來，而且會建立累積記錄檔案以校正本機電腦時鐘。使用累積記錄檔案，就可以在電腦一開機後立即計算硬體時鐘的系統錯誤。它會立即使用校正，使系統時間具有更高的穩定性。

有兩種方式可將NTP 機制做為用戶端。首先，用戶端可以定期查詢已知伺服器的時間。隨著用戶端的增加，此方法可能造成伺服器的高負載。其次，用戶端可以等待網路中的廣播時間伺服器所送出的NTP 廣播。此方法具有伺服器品質未知的缺點，而且伺服器送出錯誤的資訊可能造成嚴重的問題。

如果時間是經由廣播取得，就不需要伺服器名稱。在這樣的情形下，請在/etc/ntp.conf 組態檔中輸入broadcastclient。若要完全使用一或多個已知的時間伺服器，請輸入以servers 開頭的名稱。

28.2 設定本機參考時鐘

軟體套件xntp 包含與本機參考時鐘連線的驅動程式。支援的時鐘清單可以在/usr/share/doc/packages/xntp-doc/html/refclock.htm 檔案的xntp-doc 套件中找到。每個驅動程式都與數字關聯。在xntp 中，實際組態工作是利用虛擬IP 來執行。把時鐘當成在網路中一樣，將它輸入/etc/ntp.conf 檔案中。因此，會以127.127.t.u 的格式指定特殊的IP 位址給它們。在此，t 代表時鐘的類型並可決定使用哪一個驅動程式，而u 是代表單位，可決定使用哪一個介面。

一般而言，個別裝置都具有描述組態細節的特殊參數。`/usr/share/doc/packages/xntp-doc/html/driverNN.htm` 檔案(NN 是驅動程式的號碼) 提供關於時鐘特定類型的資訊。例如，“type 8” 時鐘(透過序列介面的收音機時鐘) 需要其他可以更精確地指定時鐘的模式。例如，Conrad DCF77 接收器模組具有模式 5。若要使用此時鐘做為偏好的參考，請指定 `prefer` 關鍵字。Conrad DCF77 接收器模組的完整 `server` 行如下所示：

```
server 127.127.8.0 mode 5 prefer
```

其他的時鐘也使用相同的模式。安裝 `xntp-doc` 套件後，就可以在 `/usr/share/doc/packages/xntp-doc/html` 目錄中找到 `xntp` 的文件。`/usr/share/doc/packages/xntp-doc/html/refclock.htm` 檔案提供描述驅動程式參數的驅動程式頁面連結。

28.3 使用YaST 設定NTP 用戶端

除了 `xntp` 的手動設定組態之外，SUSE LINUX 還可以使用 YaST 來設定 NTP 用戶端的組態。您可以使用簡便快速的組態或複雜的組態。接下來將描述這兩種組態。

28.3.1 快速NTP 用戶端組態

簡易的 NTP 用戶端組態是由兩個對話方塊組成。請在第一個對話方塊中設定 `xntpd` 的開始模式以及要查詢的伺服器。當系統開機時，若要自動啓動 `xntpd`，請按一下‘開機系統時’。然後按一下‘選取’ 存取第二個對話方塊，並在其中為網路選取適當的時間伺服器。

在詳細的伺服器選取對話方塊中，請決定執行時間同步化時，是要使用區域網路中的時間伺服器，還是使用可以處理時區的網際網路時間伺服器(‘公用 NTP 伺服器’)。若需本機時間伺服器，請按一下‘查詢’，開始 SLP 查詢，在網路中尋找可用的時間伺服器。從搜尋結果清單中選取最合適的時間伺服器，並按一下‘確定’，結束對話方塊。若需公用時間伺服器，請選取您的國家(時區) 並從‘公用 NTP 伺服器’ 下的清單選取適當的伺服器，然後按一下‘確定’，結束對話方塊。在主對話方塊中，使用‘測試’ 來測試所選取伺服器的可用性，並按一下‘完成’，結束對話方塊。

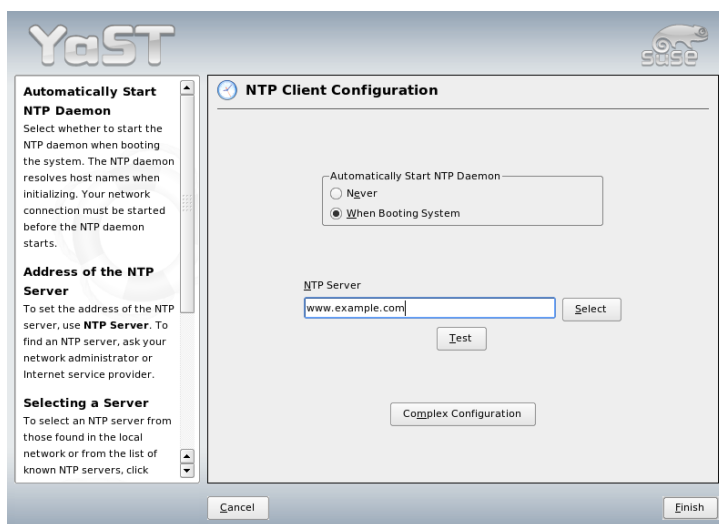


Figure 28.1: YaST：設定NTP用戶端

28.3.2 複雜的NTP用戶端組態

依照快速組態所述選取了啟動模式後，即可從‘NTP用戶端’模組主對話方塊中的‘複雜組態’下，存取NTP用戶端的複雜組態，如圖形28.1, “YaST：設定NTP用戶端”所顯示。

在‘複雜NTP用戶端組態’中，決定是否要在chroot jail中啟動xntpd。因為它可以防止攻擊者損毀整個系統，因此在xntpd遭受攻擊時，會有較高的安全性。‘透過DHCP設定NTP精靈’可以設定NTP用戶端，而能透過DHCP取得網路中可用的NTP伺服器清單。

用戶端要查詢的伺服器以及其他時間來源會列在清單下半部。修改清單時，可依需要使用‘新增’、‘編輯’以及‘刪除’。‘進階’可以讓您檢視用戶端的日誌檔，也可視NTP用戶端的組態來調整防火牆。

按一下‘新增’以新增時間資訊的新來源。在下列對話方塊中，選取進行時間同步化的來源類型。下列是可用的選項：

伺服器 另一個對話方塊可讓您選取NTP伺服器(如節28.3.1, “快速NTP用戶端組態”中所述)。在系統開機時，請啟用‘用以啓始同步化’來觸發伺服器與用

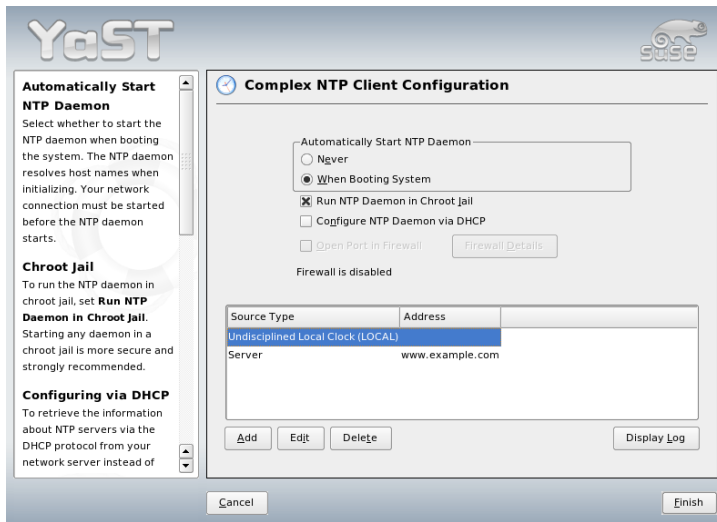


Figure 28.2: YaST：複雜的NTP用戶端組態

用戶端之間的時間資訊同步化。您可以在輸入欄位指定xntpd的其他選項。如需詳細資訊，請參閱 `/usr/share/doc/packages/xntp-doc`。

點 「點」是指建立對稱關係的機器：它可同時做為時間伺服器及用戶端。若要在相同的網路中使用點而非伺服器，請輸入個別系統的位址。其餘的對話方塊與‘伺服器’對話方塊相同。

收音機時鐘 若要在系統中使用收音機時鐘來進行時間同步化，請在此對話方塊中輸入時鐘類型、單位編號、裝置名稱以及其他選項。按一下‘裝置校正’，即可微調裝置。如需關於本機收音機時鐘作業的詳細資訊，可在 `/usr/share/doc/packages/xntp-doc/html/refclock.htm` 中找到。

廣播 時間資訊與查詢也可以透過網路中的廣播傳輸。請在此對話方塊中輸入廣播所應傳送至的位址。除非您已經有類似收音機控制時鐘的可靠時間來源，否則請勿啟動廣播。

接受廣播封包 如果您想要讓用戶端透過廣播接收其資訊，請在這些欄位中輸入應該接受的個別封包位址。

LDAP — 一種目錄服務

「輕量型目錄存取協定」(Lightweight Directory Access Protocol, LDAP) 是通訊協定集合用以存取和維護資訊目錄。LDAP 可做為許多用途，像是使用者與群組管理、系統組態管理或是位址管理。本章針對LDAP 如何運作以及如何使用YaST 管理LDAP 資料提供基本的概念。

29.1	LDAP 與NIS 的比較	413
29.2	LDAP 目錄樹的結構	413
29.3	使用slapd.conf 來設定伺服器	416
29.4	LDAP 目錄中的資料處理	420
29.5	YaST LDAP 用戶端	424
29.6	若需詳細資訊	429

在網路環境中保持重要資訊的結構性和容易存取是相當重要的。要達到此一目的，可使用黃頁之類的目錄服務，它會以結構良好並且快速搜尋的形式來保存資訊以供存取。

在理想的狀況下，中央伺服器會將資料保存在目錄中，並使用特定的協定將資料分送給所有用戶端。資料的結構允許各種不同的應用程式進行存取。如此，個別的行事曆工具和電子郵件用戶端就不需要維護自己的資料庫，而是向中央儲存區存取所需的資料。此一特性明顯降低管理資訊的需要。使用LDAP等開放且標準化的協定，以盡量確保所有不同的用戶端應用程式都能存取這類的資訊。

此處所指的目錄是一種最佳化的資料庫，可供快速而有效的讀取和搜尋：

- 爲了能夠同時大量讀取，寫入權限僅限管理員所執行的少量更新作業。舊式的資料庫會進行最佳化，以容許在最短時間內接受最多的資料量。
- 由於寫入的存取限制極大，因此採用目錄服務來管理大多數未變更的靜態資料。舊式資料庫中的資料經常變更(動態資料)。舉例來說，企業目錄中的電話號碼變更頻率，會低於會計部門所管理數字的變更頻率。
- 管理靜態資料時，很少會更新現存的資料集。管理動態資料時，特別是銀行帳戶或會計相關的資料集時，最重要的是保持這些資料的一致性。如果要從一筆資料扣除一個數目，然後將它加到另一筆資料，則必須在同一筆交易內同時進行這兩個動作，以確保整個資料集的餘額正確無誤。資料庫可支援此種交易。目錄則否。目錄可接受短暫性的資料不一致。

LDAP 這類的目錄服務並不是設計來支援複雜的更新或查詢。所有的應用程式在存取這種服務時，都應該以快速簡便的方式進行。

許多目錄先前即已存在，而且現在仍存在於Unix內和Unix外。Novell NDS、Microsoft ADS、Banyan的Street Talk和OSI標準的X.500是其中的幾個例子。LDAP原先的設計只是DAP的簡易版。DAP是一個目錄存取協定，可用來存取X.500。X.500標準負責管理目錄項目的階層組織。

LDAP是DAP的精簡版。在不失去X.500項目階層的前提下，擁有LDAP跨平台功能並節省資源。由於使用TCP/IP，使得在擴充應用程式和LDAP服務之間建立介面的工作變得更加容易。

然而LDAP的功能已經提昇。在愈來愈多的使用中，LDAP已不需要X.500的支援而可作為獨立的解決方案使用。LDAP支援使用LDAPv3 (openldap2套件中的協定版本)的轉介，此一特性使它具備實作分散式資料庫的能力。SASL(簡單驗證及安全性階層)的使用也是一項新嘗試。

LDAP超越其原始設計，其用途不限於查詢X.500伺服器的資料。slapd開放來源伺服器可將物件資訊儲存在本地資料庫。另外有一個稱為slurpd的延伸套件，負責複製多個LDAP伺服器。

openldap2 套件包含：

`slapd` 是一個獨立的LDAPv3 伺服器，用來管理BerkeleyDB 架構資料庫中的物件資訊。

`slurpd` 此程式可將本地LDAP 伺服器中的資料變更複製到其它安裝在網路中的LDAP 伺服器。

其它系統維護工具 `slapcat`、`slapadd`、`slapindex`

29.1 LDAP 與NIS 的比較

Unix 系統管理員習慣使用NIS 伺服器進行名稱解析和網路資料散佈。包含於`/etc`中的檔案和`group`、`hosts`、`mail`、`netgroup`、`networks`、`passwd`、`printcap`、`protocols`、`rpc` 和`services` 等目錄下的組態資料是由網路中的用戶端進行散發。因為這些檔案是簡單的文字檔，因此相當容易維護。不過，因為缺乏結構，較大量的資料處理變得更加困難。NIS 的設計只適用於Unix 平台，因此不可能作為異質性網路的中央資料管理員。

和NIS 不同，LDAP 伺服器不限於純Unix 網路。Windows 伺服器(2000 或更新)支援LDAP 的目錄服務功能。Novell 也提供LDAP 服務。其它非Unix 系統也可為前述應用程式工作提供更多支援。

LDAP 規則可套用於任何必須進行中央管理的資料結構。以下為幾個應用程式範例：

- 作為NIS 服務的替代品使用
- 郵件路由(`postfix`、`sendmail`)
- 郵件用戶端的通訊錄，例如Mozilla、Evolution 和Outlook。
- BIND9 名稱伺服器的區域描述管理

這個清單還可加長，因為LDAP 是可以延伸的，和NIS 有所不同。由於清楚定義的資料階層結構可方便搜尋，使得管理大量資料的工作更為容易。

29.2 LDAP 目錄樹的結構

LDAP 的目錄具有樹狀結構。階層中的所有目錄項目(稱為物件)都有一個規定的位置。此階層稱為*目錄資訊樹*或簡稱DIT。通往所需項目的完整路徑稱為*可辨識名稱*或DN，可用來清楚辨識路徑。此項目路徑中的單一節點稱為*相關可辨識名稱*或RDN。可將物件指定為下列兩種類型：

容器 這些物件可包含其它物件。這些物件的類別為root (目錄樹的根元素，此元素其實並不存在)、c (國家)、ou (組織單位) 和dc (領域元件)。此模型可媲美檔案系統中的目錄(資料夾)。

分葉 這些物件位於分支末端，而且沒有附屬物件。例如person、inetOrgPerson 或groupofNames。

目錄階層的頂端有一個根元素root。其中可包含c (國家)、dc (領域元件) 或o (組織) 作為其附屬元件。從下列範例(顯示於圖形29.1, "LDAP 目錄結構") 可清楚看出LDAP 目錄樹中的關係。

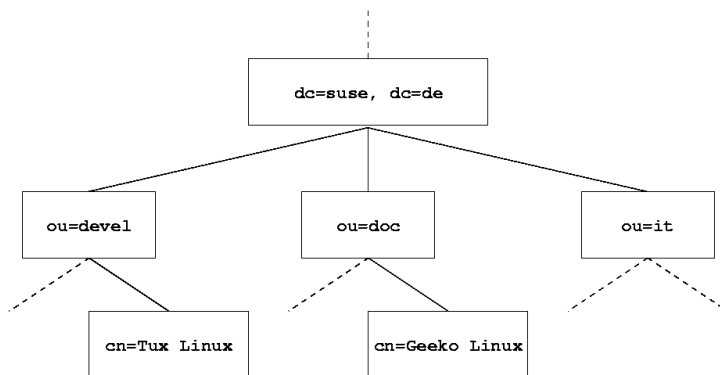


Figure 29.1: LDAP 目錄結構

這個完整的結構圖是由一棵虛擬的目錄資訊樹所組成。其中包含三個階層的項目。每個項目分別對應到圖上的一個方塊。在此範例中，虛擬的 SUSE 員工 Geeko Linux 有一個完整有效的可辨識名稱，即 `cn=Geeko Linux, ou=doc, dc=suse, dc=de`。此名稱的形成是將 RDN `cn=Geeko Linux` 加到前一個項目 `ou=doc, dc=suse, dc=de` 的 DN。

哪一種類型的物件應儲存在 DIT 的全域決定，是根據綱要而定的。物件類型則是由物件類別來決定。物件類別決定相關物件必須或可以被指定哪一種屬性。因此，綱要的內容必須包括所有物件類別的定義，以及所需應用程式案例中使用的屬性。有幾個通用綱要(請參閱 RFC 2252 和 2256)。不過，仍可建立自訂的綱要。如果用來操作 LDAP 的伺服器環境需要，也可使用多個綱要來互相輔助。

表格 29.1, "常用物件類別和屬性" 提供一個小型綜覽，介紹範例中所使用的 `core.schema` 和 `core.schema` 物件類別，包括所需的屬性和有效的屬性值。

Table 29.1: 常用物件類別和屬性

物件類別	意義	範例項目	強制屬性
dcObject	<i>domainComponent</i> (領域的名稱元件)	suse	dc
organizationalUnit	<i>organizationalUnit</i> (組織單位)	doc	ou
inetOrgPerson	<i>inetOrgPerson</i> (內部網路或網際網路的相關人員資料)	Geeko Linux	sn 與 cn

範例29.1, “摘自schema.core (行列編號為說明之用)” 中摘述一段綱要指示詞及其說明。

Example 29.1: 摘自schema.core (行列編號為說明之用)

```
#1 attributetype (2.5.4.11 NAME ( 'ou' 'organizationalUnitName' )
#2     DESC 'RFC2256: organizational unit this object belongs to'
#3     SUP name )
...
#4 objectclass ( 2.5.6.5 NAME 'organizationalUnit'
#5     DESC 'RFC2256: an organizational unit'
#6     SUP top STRUCTURAL
#7     MUST ou
#8 MAY (userPassword $ searchGuide $ seeAlso $ businessCategory
    $ x121Address $ registeredAddress $ destinationIndicator
    $ preferredDeliveryMethod $ telexNumber
    $ teletexTerminalIdentifier $ telephoneNumber
    $ internationaliSDNNumber $ facsimileTelephoneNumber
    $ street $ postOfficeBox $ postalCode $ postalAddress
    $ physicalDeliveryOfficeName
    $ st $ l $ description) )
...
```

`organizationalUnitName` 屬性類型和對應的 `organizationalUnit` 物件類別在此作為範例之用。第1行的重點為屬性名稱、其唯一的OID (物件識別碼) (數值) 和屬性的縮寫。

第2行提供屬性說明和DESC。另外還包括對應的RFC，可據此列出相關定義。第3行的SUP表示此屬性所隸屬的上級屬性類型。

第4行開始為 `organizationalUnit` 物件類型的定義，內容包括OID和物件類別名稱，如同屬性定義一樣。第5行為物件類別的簡短說明。第6行的SUP top

項目表示此物件類別不隸屬於其它物件類別。第7行始於MUST，列出所有必須和organizationalUnit 類型物件配合使用的屬性類型。第8行始於MAY，列出所有允許和此物件類別配合使用的屬性類型。

有關綱要的使用，可在OpenLDAP 文件中找到一篇很好的介紹。安裝好之後，請至 `/usr/share/doc/packages/openldap2/admin-guide/index.html` 尋找這篇文章。

29.3 使用slapd.conf 來設定伺服器

在您安裝好的系統的 `/etc/openldap/slapd.conf` 中有一個完整的組態檔，可供您的LDAP 伺服器使用。此處有這個單一項目的簡介，對於必要的調整也有說明。開頭為井字符號(#) 的項目為非作用中的項目。必須移除這個備註字元，才能啟用這些項目。

29.3.1 slapd.conf 中的全域指示詞

Example 29.2: slapd.conf: 包括綱要指示詞

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/inetorgperson.schema
```

在範例29.2, “slapd.conf: 包括綱要指示詞” 中，slapd.conf 的第一個指示詞指定一個綱要，LDAP 目錄會根據此綱要來組織其架構。core.schema 項目為強制性。其它規定的綱要也已附加在指示詞中(inetorgperson.schema 已附加於此作為範例)。在 `/etc/openldap/schema` 目錄中可找到更多可用的綱要。另外包含的 `rfc2307.schema` 和 `cosine.schema` 兩個綱要可用來將NIS取代為類似的LDAP 服務。可在包含的OpenLDAP 文件中找到更多資訊。

Example 29.3: slapd.conf: pidfile 和argsfile

```
pidfile /var/run/slapd/slapd.pid
argsfile /var/run/slapd/slapd.args
```

這兩個檔案包含PID (處理程序ID) 和一些用來啓動slapd 程序的引數。此處不需要修改。

Example 29.4: slapd.conf:存取控制

```
# Sample Access Control
#     Allow read access of root DSE
# Allow self write access
#     Allow authenticated users read access
#     Allow anonymous users to authenticate
# access to dn="" by * read
#     access to * by self write
#         by users read
#         by anonymous auth
#
# if no access controls are present, the default is:
#     Allow read by all
#
# rootdn can always write!
```

範例29.4, “slapd.conf:存取控制” 摘自slapd.conf，此檔案在伺服器上負責管制LDAP目錄的存取權。只要特定資料庫區段中沒有宣告任何自訂的規則，在slapd.conf全域區段中的設定均為有效。這些自訂規則可覆寫全域宣告。正如此處所提到的，所有使用者都有讀取目錄的權利，但只有管理員(rootdn)才可在目錄中寫入資料。LDAP中的存取控制規定是一個非常複雜的程序。下列秘訣可提供協助：

- 每一項存取規則都具有下列結構：

```
access to <what> by <who> <access>
```

- *<what>* 是一個替代符號，代表有存取權的物件或屬性。個別的目錄分支可由個別規則來明確地保護。也可使用一般表示式來處理目錄樹的區域。slapd會按照組態檔中所列的順序來評估所有規則。較一般性的規則應列在較特定的規則之後—slapd視為有效的第一條規則會受到評估，其後的所有項目則被忽視。
- *<who>* 會決定誰應獲得存取權，以進入由*<what>*所決定的區域。可使用一般表示式。第一次的配對完成後slapd會再次中止who的評估，因此必須在較一般性的規則之後列出較特定的規則。在表格29.2, “使用者群組及其存取權”中所顯示的項目是有可能的。

Table 29.2: 使用者群組及其存取權

標籤	範圍
*	沒有例外的所有使用者
anonymous	未驗證的(「匿名」)使用者
users	驗證的使用者
self	以目標物件連線的使用者
dn.regex=<regex>	符合一般表示式的所有使用者

- `<access>` 會指定存取類型。使用表格29.3, “存取類型” 中所列的選項。

Table 29.3: 存取類型

標籤	存取範圍
none	沒有存取權
auth	連絡伺服器
compare	比較物件的存取
search	搜尋過濾器的使用
read	讀取權
write	寫入權

slapd 會比較用戶端所要求的存取權與在slapd.conf 中授予的存取權。如果用戶端所要求的權利比規則中的等級更低或同等，則可獲得存取權。如果用戶端要求的權利比規則中的宣告更高，則會被拒絕。

範例29.5, “slapd.conf:存取控制範例” 顯示使用一般表示式即可任意開發簡單存取控制權的範例。

Example 29.5: slapd.conf:存取控制範例

```
access to dn.regex="ou=([^\,]+),dc=suse,dc=de"
by dn.regex="cn=administrator,ou=$1,dc=suse,dc=de" write
by user read
by * none
```

此規則宣告只有個別的管理員有寫入個別ou項目的權利。其它所有通過驗證的使用者都有讀取權，其餘的人則沒有存取權。

建立存取規則

如果沒有access to的規則或者沒有符合by的指示詞，則無法取得存取權。只會授予明確宣告的存取權。如果沒有任何宣告的規則，則管理員依照預設原則擁有寫入權，其他人則有讀取權。

請至已安裝的openldap2套件的線上文件尋找詳細資訊及LDAP存取權的組態範例。

除了使用(slapd.conf)中央組態檔來管理存取許可權之外，也可使用ACI(存取控制資訊)。ACI允許在LDAP樹中儲存個別物件的存取資訊。此種類型的存取控制尚未普遍，開發者目前仍將它視為一項實驗。請參閱<http://www.openldap.org/faq/data/cache/758.html>以取得更多資訊。

29.3.2 slapd.conf 中的資料庫特定指示詞

Example 29.6: slapd.conf: 資料庫特定指示詞

```
database ldbm
suffix "dc=suse,dc=de"
rootdn "cn=admin,dc=suse,dc=de"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap
# Indices to maintain
index objectClass eq
```

資料庫類型是在本區段中的第一行中決定，在本例中為LDBM(請參閱範例29.6, "slapd.conf: 資料庫特定指示詞")。第二行使用suffix來決定本伺服器應該負責LDAP樹的哪一個部份。接下來的rootdn決定擁有本伺服器管理權的人。此處宣告的使用者不需要有LDAP項目或以一般使用者的身份出現。管理員密碼是使用rootpw來設定。在這裡可以不使用secret而是以雜湊值來輸

入slappasswd 所建立的管理員密碼。directory 指示詞表示資料庫目錄所在的目錄(在檔案系統中)儲存在伺服器中。最後一個指示詞index objectClass eq 會啟動所有物件類別索引的維護作業。在此處可根據經驗，加入一個使用者最常搜尋的屬性。此處使用的是資料庫自訂的Access 規則，而非全域的Access 規則。

29.3.3 啟動及停止伺服器

一旦LDAP 伺服器完全設定好，且所有需要的項目已依照節29.4, “LDAP 目錄中的資料處理”中描述的樣式建立之後，輸入rcldap start 以root 的身份來啟動LDAP 伺服器。輸入rcldap stop 指令可手動停止伺服器。使用rcldap status 來請求執行中LDAP 伺服器的狀態。

節7.6, “Systems Servcices (RunLevel)”中所描述的YaST runlevel 編輯器，可讓系統在啟動和暫停時自動啟動及停止伺服器。也可以透過節7.5.1, “新增init 程序檔”中所述的指令提示，利用insserv 指令來建立對應的連結至啟動和停止程序檔。

29.4 LDAP 目錄中的資料處理

OpenLDAP 提供一系列的工​​具，用來管理LDAP 目錄中的資料。以下說明四種最重要的工具，分別用來新增、刪除、搜尋和修改資料集。

29.4.1 將資料加入LDAP 目錄

一旦您LDAP 伺服器/etc/openldap/lsapd.conf 中的組態正確並且就緒(即擁有正確的suffix、directory、rootdn、rootpw 和index)項目，請接著輸入記錄。OpenLDAP 提供這項工作所需的ldapadd 指令。可能的話，將多個物件一次加入資料庫是較實際的作法。LDAP可處理此處的LDIF 格式(LDAP 資料交換格式)。LDIF 是一個簡單的文字檔，可包含任意的屬性及數值組合。有關可用的物件類別和屬性的資料，請參閱slapd.conf 中宣告的綱要檔。在圖形29.1, “LDAP 目錄結構”中用來建立概略架構的LDIF 檔會和範例29.7, “LDIF 檔案範例”中的檔案相似。

Example 29.7: LDIF 檔案範例

```
# The SUSE Organization
dn: dc=suse,dc=de
objectClass: dcObject
objectClass: organization
o: SUSE AG dc: suse

# The organizational unit development (devel)
dn: ou=devel,dc=suse,dc=de
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=suse,dc=de
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=suse,dc=de
objectClass: organizationalUnit
ou: it
```

LDIF 檔的編碼

LDAP 可使用 UTF-8 (Unicode)。母音字元的編碼必須正確。使用支援 UTF-8 的編輯器 (例如 Kate 或最新版的 Emacs)。否則，應避免母音字元和其它特殊字元，或使用 `recode` 將輸入重新編碼為 UTF-8。

儲存字尾為 `.ldif` 的檔案，然後用下列指令傳送給伺服器。

```
ldapadd -x -D <dn of the administrator> -W -f <file>.ldif
```

`-x` 在此例中會關閉 SASL 的驗證。`-D` 會宣告呼叫操作的使用者。此處輸入管理員的有效 DN，如同 `slapd.conf` 中的設定一樣。在此範例中，有效 DN 為 `cn=admin,dc=suse,dc=de`。`-W` 可避免將密碼輸入指令行 (以非加密文字) 並啟動一個密碼提示。此密碼是先前在 `slapd.conf` 中使用 `rootpw` 決定的。`-f` 傳送檔案名稱。請參閱範例 29.8, “`ldapadd` 和 `example.ldif`” 中執行 `ldapadd` 的詳細內容。

Example 29.8: ldapadd 和 example.ldif

```
ldapadd -x -D cn=admin,dc=suse,dc=de -W -f example.ldif
```

```
Enter LDAP password:  
adding new entry "dc=suse,dc=de"  
adding new entry "ou=devel,dc=suse,dc=de"  
adding new entry "ou=doc,dc=suse,dc=de"  
adding new entry "ou=it,dc=suse,dc=de"
```

個別的使用者資料可建立在不同的LDIF 檔。範例29.9, “Tux 的LDIF 資料” 會新增Tux 至新的LDAP 目錄。

Example 29.9: Tux 的LDIF 資料

```
# coworker Tux  
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de  
objectClass: inetOrgPerson  
cn: Tux Linux  
givenName: Tux  
sn: Linux  
mail: tux@suse.de  
uid: tux  
telephoneNumber: +49 1234 567-8
```

LDIF 檔可包含任意數量的物件。可以將整個目錄分支一起傳給伺服器，也可只傳送一部份，如個別物件範例所示。如果需要經常修改某些資料，建議使用單一物件的細分。

29.4.2 修改LDAP 目錄中的資料

工具`ldapmodify` 可用來修改資料集。最簡單的作法是先修改對應的LDIF 檔，然後將修改過的檔案傳送給LDAP 伺服器。如果要將同事Tux 的電話號碼從+49 1234 567-8 改為+49 1234 567-10，則必須依照範例29.10, “修改過的LDIF 檔tux.ldif” 中的方式來編輯LDIF 檔案。

Example 29.10: 修改過的LDIF 檔tux.ldif

```
# coworker Tux  
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de  
changetype: modify  
replace: telephoneNumber  
telephoneNumber: +49 1234 567-10
```


使用下列指令將修改過的檔案輸入LDAP目錄：

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W -f tux.ldif
```

或者，也可以直接將要修改的屬性傳送給`ldapmodify`。此項作業程序描述如下：

1. 啓動`ldapmodify`並輸入您的密碼：

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W
Enter LDAP password:
```

2. 輸入變更並注意是否符合下列語法順序：

```
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

有關`ldapmodify`及其語法的詳細資訊，請參閱對應的man頁面(`ldapmodify(1)`)。

29.4.3 搜尋或讀取LDAP目錄中的資料

OpenLDAP透過`ldapsearch`提供一個指令行工具，用來搜尋和讀取LDAP目錄中的資料。下列為簡易查詢的語法：

```
ldapsearch -x -b dc=suse,dc=de "(objectClass=*)"
```

選項`-b`決定搜尋基礎——用來執行搜尋的目錄樹區段。在目前這個範例中，此搜尋基礎為`dc=suse,dc=de`。如果要在LDAP目錄的次區段中執行更詳細的搜尋(例如，只在`devel`部門搜尋)可使用`-b`將此區段傳送至`ldapsearch`。`-x`會要求啓動簡單的驗證。`(objectClass=*)`會宣告目錄中的所有物件都應該讀取。在建立一個新的目錄樹後，可使用本指令來確認所有項目都已正確記錄，而且伺服器的回覆也符合需要。更多有關使用`ldapsearch`的資訊，可在對應的man頁面(`ldapsearch(1)`)找到。

29.4.4 刪除LDAP目錄中的資料

使用`ldapdelete`來刪除不要的資料。其語法和前述指令相似。例如，要刪除Tux Linux的整個項目：可發出下列指令：

```
ldapdelete -x -D cn=admin,dc=suse,dc=de -W cn=Tux \
Linux,ou=devel,dc=suse,dc=de
```

29.5 YaST LDAP 用戶端

YaST 中包含一個模組，可用來設定LDAP式的使用者管理。如果您在安裝時未啟用這項功能，可選取‘網路服務’→‘LDAP 用戶端’來啓用它。YaST 會根據LDAP (如下所述) 的要求自動啓用PAM 和NSS相關的變更，並安裝所需的檔案。

29.5.1 標準程序

必須知道用戶端機器的處理程序，以了解YaST LDAP 用戶端模組的作業。如果已啓動LDAP 以進行網路驗證或者已呼叫YaST 模組，則表示已安裝pam_ldap 和nss_ldap 套件，而且兩個對應的組態檔也已經調整。pam_ldap 為PAM 模組，負責登入程序和LDAP 目錄之間的協調，也是驗證資料的來源。已安裝pam_ldap.so 專用模組並調整PAM 組態(請參閱範例29.11, “適用於LDAP 的pam_unix2.conf”)。

Example 29.11: 適用於LDAP 的pam_unix2.conf

```
auth:         use_ldap nullok
account:      use_ldap
password:     use_ldap nullok
session:      none
```

在手動設定額外服務以使用LDAP 時，應納入PAM組態檔中的PAM LDAP 模組，該組態檔與/etc/pam.d 中的服務相對應。可在/usr/share/doc/packages/pam_ldap/pam.d/ 中找到適用於個別服務的組態檔。將適當的檔案複製到/etc/pam.d 。

透過nsswitch 機制進行的glibc 名稱解析適用於含有nss_ldap 的LDAP。安裝此套件時，已在/etc/ 中建立一個新的調整過的nsswitch.conf 檔案。更多有關nsswitch.conf 的作業可在節22.5.1, “組態檔” 中找到。nsswitch.conf 必須包含下列幾行，以供LDAP 進行使用者管理及驗證。請參閱範例29.12, “nsswitch.conf中的調整”。

Example 29.12: nsswitch.conf中的調整

```
passwd: compat
group: compat

passwd_compat: ldap
group_compat: ldap
```

這幾行會先命令glibc的查詢程式庫評估/etc中的對應檔案，然後存取LDAP伺服器，以它作為驗證及使用者資料的來源。可對此機制進行測試，例如使用getent passwd指令來讀取使用者資料庫的內容。所傳回的資料集應包含一份有關您系統上的本地使用者和LDAP伺服器儲存的所有使用者的調查。

如果要避免一般使用者透過LDAP以ssh或login登入伺服器，則/etc/passwd和/etc/group檔案必須各自增加一行。該行的內容為+:::/:sbin/nologin in /etc/passwd and +:::in /etc/group。

29.5.2 LDAP 用戶端的組態

當YaST依照要求的方式修改nss_ldap、pam_ldap、/etc/passwd和/etc/group之後，即可在YaST對話方塊中開始真正的組態設定作業。請參閱圖形29.2, “YaST: LDAP用戶端的組態”。

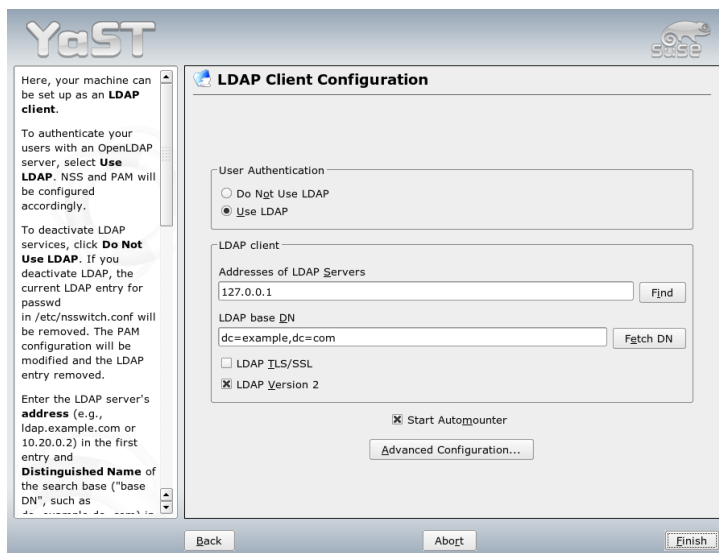


Figure 29.2: YaST: LDAP 用戶端的組態

啓用LDAP以進行第一個對話方塊中的使用者驗證。輸入伺服器中的搜尋基礎，其下的所有LDAP伺服器資料都儲存在‘LDAP基礎DN’中。輸入可在‘LDAP伺服器位址’中找到的LDAP伺服器位址。如果要讓目錄自動裝置在遠端主機，請

選取‘啓動自動裝載器’。如果要以管理者身份修改伺服器上的資料，按一下‘進階組態’。請參閱圖形29.3, “YaST:進階組態”。

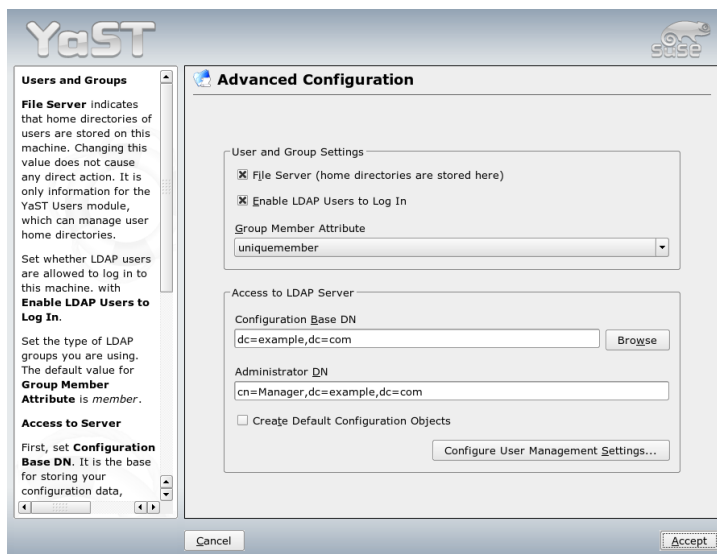


Figure 29.3: YaST:進階組態

下一個對話方塊包括兩個部份：上半部為使用者和群組可用的一般選項，如YaST使用者模組所反映的一樣。下半部為存取LDAP伺服器所需的資料。使用者和群組設定包含下列項目：

檔案伺服器 如果現用的系統為檔案伺服器，並使用 /home 來包含所有使用者目錄，則啓用此選項可確保YaST模組以正確方式處理使用者目錄。

允許LDAP使用者登入 啓用此選項之後，透過LDAP管理的使用者即可獲得登入系統的許可權。

群組成員屬性 此選項可指定要使用的LDAP群組類型，‘成員’（預設值）或‘獨特成員’。

輸入所需的存取資料，以在此修改LDAP伺服器上的組態。這些組態為‘組態基礎DN’，所有組態物件均儲存於其下，另外還有‘管理員DN’。

按一下‘設定使用者管理設定’以編輯LDAP 伺服器上的項目。將您的LDAP密碼輸入顯示的對話方塊中，以進行伺服器驗證。接著會根據伺服器中儲存的ACL和ACI來授予伺服器組態模組的權限。

使用YaST 用戶端

使用YaST LDAP 用戶端來調整YaST 模組，以進行使用者和群組管理，並在需要時加以延伸。另外，還可用預設值來定義個別屬性樣板，以簡化實際的資料註冊。此處所建立的預設會以LDAP 物件的形式儲存在LDAP 目錄中。使用者資料註冊仍以一般YaST 模組輸入形式來完成。已註冊的資料會被當成物件儲存在LDAP 目錄。

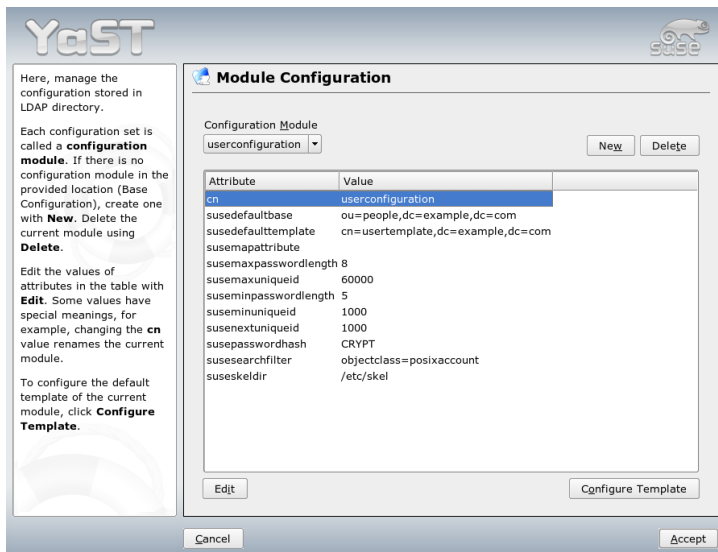


Figure 29.4: YaST: 模組組態

模組組態對話方塊(圖形29.4, “YaST: 模組組態”)允許選取和修改現有的模組組態、建立新模組以及設計和修改這些模組的樣板。如果要修改組態模組中的值，或重新命名模組，請從現有模組的內容上方選取一個模組類型。接著該內容會顯示一個表格，列出所有可用於此模組的屬性和其指定值。除了所有設定的屬性之

外，表格中也會列出現用綱要允許但未使用的其它屬性。

如果要複製模組，只要變更cn即可。如果要修改個別的屬性值，選取該屬性然後按一下‘編輯’。會出現一個對話方塊，可在其中變更所有該屬性的設定。按‘確定’確認變更。

如果要將新模組加入現有的模組，按一下內容上方的‘新增’。在出現的對話方塊中輸入新模組的名稱和物件類別(suseuserconfiguration或susegroupconfiguration)。按‘確定’關閉對話方塊後，新模組即新增至現有模組的選單中，可加以選取或取消選取。按一下‘刪除’可刪除目前選取的模組。

如果先前已使用YaST LDAP 用戶端加以定義，則用來管理使用者和群組的YaST 模組會包含已設定合理標準值的樣板。如果要編輯樣板，按一下‘設定樣板組態’。下拉式功能表包含可修改的現有樣板或空白項目。選擇其中之一，然後在‘物件樣板組態’表格中設定此樣板內容(請參閱圖形29.5, “YaST：物件樣板組態”)。這個表格分為兩個表格式的綜覽視窗。上方視窗列出所有一般樣板屬性。請依照您的需要來決定適當的值，或者將一些項目保留空白。LDAP 伺服器上的空白屬性會被刪除。

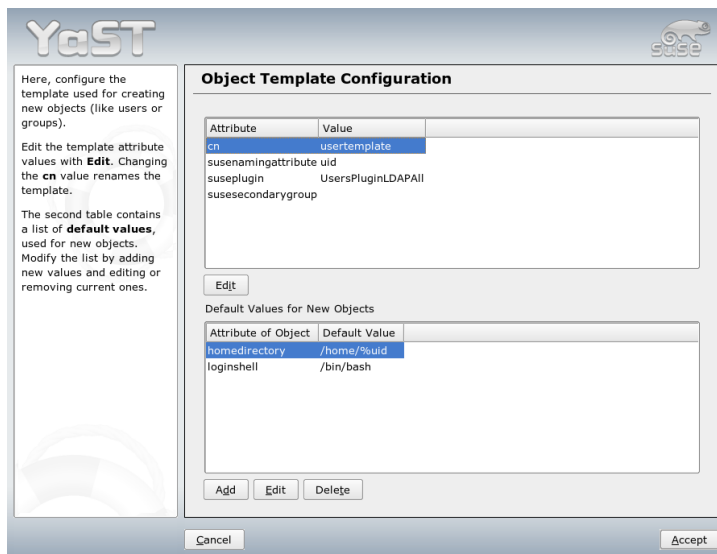


Figure 29.5: YaST：物件樣板組態

第二個綜覽視窗(‘新物件的預設值’)列出所有對應的LDAP物件(在此為群組或使用者組態)，各個物件均定義一個標準值。可加入其它屬性標準值，可編輯現有屬性和其設定值，還可刪除整個屬性。可透過變更cn項目來複製樣板。如前所述，將模組的susedefaulttemplate屬性值設定為調整過的樣板DN，以連接樣板與其模組。

使用變數樣式來取代絕對值，即可透過其它屬性建立一個屬性的預設值。例如，建立新使用者時，會自動以sn和givenName的屬性值建立cn=%sn %givenName。

一旦所有模組和樣板都已正確設定並可執行時，可使用YaST以一般方式來註冊新群組和使用者。

29.5.3 使用者和群組—使用YaST設定組態

實際的使用者和群組資料註冊與不使用LDAP時的程序只有些微不同。以下是使用者管理簡介。群組管理程序與此類似。

經由‘安全性與使用者’→‘使用者管理’進入YaST使用者管理。會顯示一個表格，用來註冊最重要的使用者資料，例如姓名、登入設定和密碼。透過‘詳細資訊’可存取群組成員的組態表、登入外圍程序和主目錄。預設值是依照節29.5.2, “LDAP用戶端的組態”中所描述的程序來定義。使用LDAP時，可由此表格進入另一個用來註冊LDAP特定屬性的表格。如圖形29.6, “YaST：其它LDAP設定”所示。選取所有要變更數值的屬性，然後按一下‘編輯’。關閉使用‘繼續’開啓的表格，即可回到最初的使用者管理輸入表。

最初的使用者管理輸入表中包含‘LDAP選項’。此功能可用來將LDAP搜尋過濾器套用到可用的使用者，或者也可選取‘LDAP使用者及群組組態’來連接設定LDAP使用者和群組的模組。

29.6 若需詳細資訊

本章節刻意將一些較為複雜的主題排除在外，例如SASL組態或分散工作量至多個從屬伺服器的LDAP伺服器的複製。有關這兩個主題的詳細資訊，可在*OpenLDAP 2.2 管理員指南*中找到(請參閱下列資訊)。

OpenLDAP專案的網站提供LDAP初學者及進階使用者詳盡的說明。

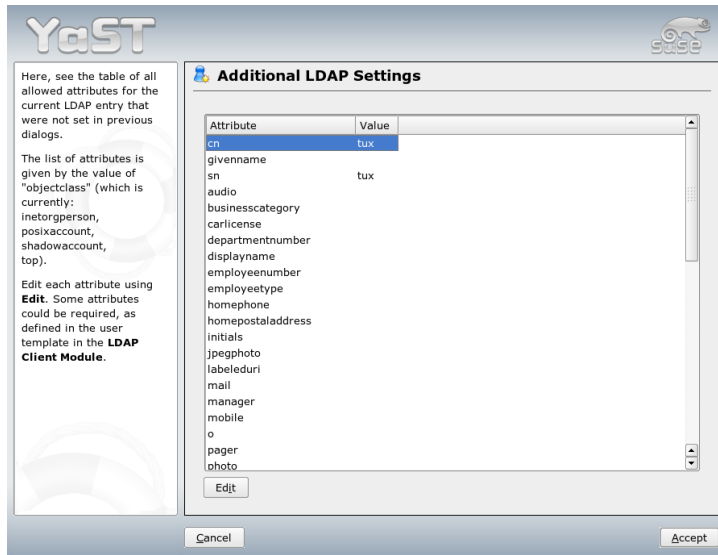


Figure 29.6: YaST：其它LDAP 設定

OpenLDAP Faq-O-Matic 是一個豐富的問答集，內容包括安裝、組態設定及OpenLDAP 的使用。http://www.openldap.org/faq/data/cache/1.html.

快速入門指南 簡潔的逐步示範，教您如何安裝您的第一部LDAP 伺服器。

http://www.openldap.org/doc/admin22/quickstart.html
 或是在/usr/share/doc/packages/openldap2/admin-guide/quickstart.html 中已安裝的系統

OpenLDAP 2.2 管理員指南 詳細介紹所有重要的LDAP 組態，包括存取控制及加密。http://www.openldap.org/doc/admin22/ 或者在安裝系統中的/usr/share/doc/packages/openldap2/admin-guide/index.html

以下是和LDAP 有關的IBM Redbook：

了解LDAP 有關LDAP 基本原則的一般性介紹：http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf.

LDAP 實作指南 主要針對 *IBM SecureWay Directory* 管理員的需要來介紹。不過，也可在這個網站找到重要的LDAP一般資訊：<http://www.redbooks.ibm.com/redbooks/pdfs/sg245110.pdf>。

有關LDAP的出版品：

- Howes, Smith, and Good: *Understanding and Deploying LDAP Directory Services*. Addison-Wesley, 2nd ed., 2003. (ISBN 0-672-32316-8)
- Hodges: *LDAP System Administration*. O'Reilly & Associates, 2003. (ISBN 1-56592-491-6)

有關LDAP的最終參考資料為對應的RFC (要求建議)，2251 至2256。

Apache 網頁伺服器

Apache 擁有超過60%的市場佔有率，是世界上使用最普遍的網頁伺服器(來源：<http://www.netcraft.com>)。針對網頁應用程式設計的Apache，通常是由MySQL 資料庫以及PHP 和Perl 程式語言結合而成。此結合一般稱為LAMP。

本章節將介紹Apache 網頁伺服器。並且說明執行安裝和組態的方法，此外也會描述到一些可用模組，還會介紹到虛擬主機的變化。

30.1	基本操作	434
30.2	使用YaST 設定HTTP 伺服器	435
30.3	Apache 模組	435
30.4	執行緒	436
30.5	安裝	437
30.6	組態	438
30.7	使用Apache	442
30.8	主動內容	443
30.9	虛擬主機	447
30.10	安全性	450
30.11	疑難排解	451
30.12	若需詳細資訊	451

30.1 基本操作

本節將提供網頁伺服器及其使用協定的基本介紹，還會介紹多項重要功能。

30.1.1 網頁伺服器

網頁伺服器會發出用戶端要求的HTML網頁。這些網頁可以儲存在目錄中(被動或靜態網頁)，或因應查詢而產生(主動內容)。

30.1.2 HTTP

用戶端通常是網路瀏覽器，例如Konqueror或Mozilla。瀏覽器和網頁伺服器之間的通訊，會使用超文字傳輸協定(HTTP)進行。目前的HTTP 1.1版本已記載於RFC 2068和更新的RFC 2616。這些RFC可以從<http://www.w3.org>取得。

30.1.3 URL

用戶端使用URL，例如<http://www.novell.com/linux/suse/>，向伺服器索取網頁。URL包含：

協定 常用的協定：

- `http://` HTTP 協定
- `https://` HTTP 安全加密版
- `ftp://` 下載及上傳檔案的傳輸協定

領域 在此範例中，就是www.suse.com。領域可以細分成兩個部份。第一個部份(www)指向電腦。第二個部份(suse.com)是實際領域。二者加在一起，稱為FQDN(完整領域名稱)。

資源 在此範例中，index_us.html。此部份指定前往資源的完整路徑。資源可以是檔案，和此範例一樣。然而，它可能也是CGI程序檔、JavaServer頁面，或是一些其他資源。

負責任的網際網路機制，例如領域名稱系統(DNS)，可以將查詢傳送至領域www.suse.com，引導它至一或多個裝載資源的電腦。接著Apache會從它的檔案目錄傳送實際的資源，此範例中是指index_us.html網頁。在

此狀況下，檔案位於目錄的最上層。不過，資源也可以位於子目錄中，例如 `http://support.novell.com/linux/`。

檔案路徑和 `DocumentRoot` 是相對的，可以在組態檔中變更。節 30.6.2, “`DocumentRoot`” 敘述完成的方法。

30.1.4 自動顯示預設網頁

如果沒有指定預設頁面，Apache 會自動將一個一般名稱附加到 URL。類似網頁最常用的名稱是 `index.html`。如節 30.6.2, “`DirectoryIndex`” 所述，此功能可以和伺服器應該使用的實際網頁名稱一起設定。在此範例中，`http://www.suse.com` 足以提示伺服器傳遞網頁 `http://www.novell.com/linux/suse/`。

30.2 使用 YaST 設定 HTTP 伺服器

透過 YaST 的協助，可以輕鬆地設定 Apache，但是使用此種方式設定網頁伺服器，需要對此主題具有基本了解。選取 YaST 控制中心的‘網路服務’→‘HTTP 伺服器’後，會提示您需要安裝部份缺漏的套件。每個程式安裝完成時，YaST 會顯示組態對話方塊（‘HTTP 伺服器組態’）。

請在本對話框中，先開啓‘HTTP 服務’本身。這樣也會開啓防火牆中的對應埠（埠 80）（‘開啓選取埠上的防火牆’）。視窗的下方（‘設定/摘要’）允許設定本地 HTTP 伺服器：‘監聽’（預設值是埠 80）、‘模組’、‘預設主機’以及‘主機’。‘編輯’允許變更目前選取的設定。

首先，請選取‘預設主機’，然後，如有需要，可根據您的需求來調整組態。接下來，使用‘模組’啓動想要的模組。另外有一些對話方塊，可做更詳細的組態，特別是用於建立虛擬主機。

30.3 Apache 模組

透過模組，Apache 可以使用各式各樣的函數加以擴充。例如，Apache 可以存取這些模組來執行多種程式設計語言的 CGI 程序檔。除了 Perl 和 PHP 之外，也可以使用 Python 或 Ruby 等的其他指令碼語言。也有一些模組可用於安全資料傳輸（secure sockets layer, SSL）、使用者認證、擴充的記錄以及其他功能。

若具備所需的知識，可以使用自訂撰寫的模組，依各種需要和偏好來調整 Apache。如需相關資訊，請參閱節 30.12.4, “其他來源”。

有多個“處理程式”可供指定，來處理查詢(透過組態檔中的指示)。這些處理程式可以是Apache的一部份，也可以是為處理查詢而呼叫的模組，因此安排此程序的方式很有彈性。也可以搭配自訂模組使用Apache，改變處理要求的方式。

Apache的模組化已經達到進階層次，除了小部份工作外，幾乎全部由模組來處理。這種進步幅度很大，讓HTTP透過模組的方式來處理。因此，Apache不只是網頁伺服器，它和其他模組並用，就有截然不同的用途。例如，目前有一個以Apache為基礎的證明概念郵件伺服器(POP3)。

Apache 模組提供多種其他有用的功能：

虛擬主機 支援虛擬主機，表示Apache的例項和一台機器可供多個網站使用。對於使用者而言，網頁伺服器會顯示成多個獨立的網頁伺服器。虛擬主機可以設定在不同的IP位址或名稱基礎上。這可以節省額外機器的取得成本和管理負荷。

彈性的URL重寫 Apache具備一些操控和重寫URL的方法。若需詳細資料請查看Apache文件。

內容協商 Apache可以傳送適合用戶端(瀏覽器)功能的網頁。例如，它可以為舊瀏覽器或只支援在文字模式下作業的瀏覽器，例如Lynx，提供無框架的簡單版本。如果您準備為每一種瀏覽器去調整JavaScript程式碼，上述方法就可以為每一個瀏覽器提供適用的網頁版本，避免各種瀏覽器的不相容性。

彈性錯誤處理 發生錯誤時，做出彈性反應並提供合適的回應，例如遇到網頁不存在時的狀況。回應還可以機動產生，例如透過CGI。

30.4 執行緒

執行緒是“輕量級”的處理程序。執行緒和處理程序相比的優點是，它消耗的資源較少。因此，使用執行緒取代處理，會提高效率。缺點是在執行緒環境執行的應用程式，必須具執行緒安全性。這表示：

- 函數(或物件導向應用程式中的方法)必須可重新進入—具相同輸入的函數必須永遠傳回相同結果，即使其他執行緒同時執行相同函數。因此，函數必須設計為可以由多個執行緒同時執行。
- 存取資源(通常是變數)必須安排，使並行的執行緒不會互相衝突。

Apache 2 會將查詢當成個別程序來處理，或是以結合程序或執行緒的混合模式來處理。MPM *prefork* 負責程序的執行。MPM *worker* 會提示執行緒的執行。安裝時請選取要使用的MPM (請參閱節30.5, “安裝”)。第三個模式—*perchild*—尚未完全成熟，因此無法在SUSE LINUX 進行安裝。

30.5 安裝

30.5.1 選取YaST 中的套件

對於基本安裝，選取Apache 套件`apache2` 已經足夠。此外，再安裝一個MPM (多重處理模組) 套件，例如`apache2-prefork` 或`apache2-worker`。選擇MPM 時，記住以執行緒為工作基礎的MPM，不可以用於`mod_php4`，因為有些`mod_php4` 程式庫尚未確定執行緒安全性。

30.5.2 啓動Apache

安裝之後，Apache 必須啓動成runlevel 編輯器的服務。若要在系統開機時啓動，請核取runlevel 編輯器的runlevel 3 和5。若要測試Apache 是否執行中，請將瀏覽器移至`http://localhost/`。如果Apache 正在執行，就會看到範例網頁(假設已安裝`apache2-example-pages`)。

30.5.3 主動內容的模組

若要利用模組來使用主動內容，請安裝各程式設計語言的模組。有Perl 適用的`apache2-mod_perl`，PHP 適用的`mod_php4`，以及Python 適用的`mod_python`。有關這些模組的使用，節30.8.4, “使用模組產生主動內容” 中有詳細的說明。

30.5.4 其他建議套件

建議安裝套件`apache2-doc` 提供的文件。按照節30.5.2, “啓動Apache” 敘述的方法安裝套件並啓動伺服器之後，文件可以直接由URL `http://localhost/manual` 存取。

若要開發Apache 模組或編譯協力廠商模組，除了相對應開發工具外，還需要套件`apache2-devel`。這些也包含`apxs` 工具，請參閱節30.5.5, “使用`apxs` 安裝模組”。

30.5.5 使用apxs 安裝模組

apxs2 是模組開發人員的重要工具。此程式可以使用單一指令，從原始程式碼編譯和安裝模組(包括組態檔的必要變更)。更好的是，您也可以將可用的模組安裝成物件檔案(副檔名.o)或靜態程式庫(副檔名.a)。從來源安裝模組時apxs2 會建立動態共用物件(DSO)，Apache 會直接將它當成模組使用。

使用指令，例如`apxs2 -c -i -a mod_foo.c`，從原始程式碼安裝模組。apxs2 的其他選項會在其man 頁面敘述。接著根據節30.6.1, “使用SuSEconfig 進行設定” 中的描述，以項目APACHE_MODULES 在/etc/sysconfig/apache2 中啓動模組。

apxs2 有多種版本：apxs2、apxs2-prefork 和apxs2-worker。apxs2 會安裝模組，它們就可以用於所有MPM。其他兩個程式會安裝模組，使它們只可以用於各自的MPM(prefork 或worker)。apxs2 會將模組安裝到/usr/lib/apache2，而apxs2-prefork 會將模組安裝到/usr/lib/apache2-prefork。

30.6 組態

Apache 安裝之後，僅在您有特殊需要或偏好時才需要做其他的變更。Apache 可以使用YaST 或透過直接編輯檔案/etc/apache2/httpd.conf 。

30.6.1 使用SuSEconfig 進行設定

在/etc/sysconfig/apache2 所做的設定可透過SuSEconfig 套用在Apache 組態檔案。提供的組態選項，對於大部份的案例，都應該夠用。在檔案找到的每一個變數，都會帶有一個註解，說明它的效用。

自訂組態檔

若不直接在組態檔/etc/apache2/httpd.conf 執行變更，您可以在變數APACHE_CONF_INCLUDE_FILES 的協助下，指定自己的組態檔，例如httpd.conf.local。接下來，主要組態檔會解譯檔案。在此方式下，即使檔案/etc/apache2/httpd.conf 在新安裝時被覆寫，對組態檔所做的變更也會保留。

模組

將模組的名稱包括在變數APACHE_MODULES 所指定的清單中，可以啓動使用YaST 安裝的模組。這項變數定義在檔案/etc/sysconfig/apache2 中。

旗標

`APACHE_SERVER_FLAGS` 可以用來指定旗標，以便啓動或停止組態檔的某些段落。如果組態檔中的段落包括在

```
<IfDefine someflag>
.
.
.
</IfDefine>
```

只有變數 `ACTIVE_SERVER_FLAGS` 設定個別旗標時，才會啓動它：`ACTIVE_SERVER_FLAGS = someflag`。以此方式，可以為測試目的，輕鬆啓動或停用組態檔的衆多段落。

30.6.2 手動組態

編輯組態檔 `/etc/apache2/httpd.conf`，讓無法透過 `/etc/sysconfig/apache2` 定義的設定值來使用的功能，得以啓用。以下段落敘述可以設定的一些參數。下列的排列順序就是它們在檔案中所顯示的順序。

DocumentRoot

其中一個基本設定是 `DocumentRoot`—這是 Apache 預期伺服器應該傳送網頁的目錄。預設虛擬主機是設定成 `/srv/www/htdocs`。正常情況下，這項設定不需要變更。

Timeout

指定伺服器報告要求逾時前，要等候的時間。

MaxClients

Apache 可以同時處理的用戶端數目上限。預設值是 150，不過此值可能對於使用頻率相當大的網站而言，太小了。

LoadModule

`LoadModule` 指示會指定要載入的模組。載入順序是由模組本身來決定。這些指示也會指定包含模組的檔案。

Port

指定Apache 要在什麼連接埠監聽查詢。通常是連接埠80，HTTP 的預設連接埠。正常情況下，不應該變更這個設定。讓Apache 監聽其他連接埠的一個理由，是測試新版本的網站。在此方式下，網站的作業版本仍可由預設連接埠80 存取。

另一個原因可能是您只想讓網頁供內部網路使用，因為它們包含不對外開放的資訊。為達此目的，請將連接埠設定成像8080 的值，並透過防火牆防止外界存取這個連接埠。這樣一來，外界就無法存取該伺服器。

Directory

使用這個指示來設定目錄的存取許可權和其他許可權。DocumentRoot 也有此類型的指示。變更DocumentRoot 時，務必一併變更此處指定的目錄名稱。

DirectoryIndex

此時，決定Apache 要搜尋的檔案，以完成未指定檔案的URL。預設值是index.html。例如，如果用戶端要求的URL `http://www.example.com/foo/bar`，而目錄`foo/bar` 底下的DocumentRoot 包含檔案`index.html`，則Apache 會將此網頁傳回用戶端。

AllowOverride

Apache 傳遞文件的每一個來源目錄，可以包含一個檔案來覆寫此目錄的全域存取許可權以及其他設定。這些設定會循環套用到目前的目錄以及它的子目錄，除非它們的子目錄中有其他類似檔案，覆寫這個設定值。同樣地，指定的設定的類似檔案若位於DocumentRoot，就會套用全域。類似檔案一般的名稱是`.htaccess`，但它可以變更，方法請見節30.6.2, "AccessFileName"。

使用AllowOverride 來判斷，本地檔案中指定的設定，是否會覆寫全域設定。可能的值包括None、All，以及Options、FileInfo、AuthConfig 以及Limit 的任何組合。這些值的意義詳述於Apache 文件。安全的預設值是None。

Order

這個選項決定Allow 和Deny 存取許可權設定的套用順序。預設值是：

```
Order allow, deny
```

因此，先套用的是允許存取的存取許可權，然後是拒絕存取的存取許可權。相關方法會以下列其中之一為基礎：

allow all 允許每一個存取並定義例外

deny all 拒絕每一個存取並定義例外

deny all 的範例：

```
Order deny,allow
Deny from all
Allow from example.com
Allow from 10.1.0.0/255.255.0.0
```

AccessFileName

在此，設定能夠覆寫全域存取許可權的檔案名稱，以及設定Apache傳送的目錄之其他設定值(請參閱節30.6.2, "AllowOverride")。預設值是.htaccess。

ErrorLog

指定Apache記錄錯誤訊息的檔案名稱。預設值是/var/log/httpd/errorlog。虛擬主機的錯誤訊息(請參閱節30.9, "虛擬主機")也會記錄到此檔案，除非已經在組態檔的VirtualHost段落指定特定日誌檔。

LogLevel

錯誤訊息會依據不同的嚴重程度來分類。此設定會指定錯誤訊息記錄的嚴重程度。設定記錄的程度，使超過此程度的錯誤訊息列入記錄。預設值是warn。

Alias

使用別名，指定目錄的捷徑，以便直接存取此目錄。例如，即使DocumentRoot不是設定在目錄/srv/www/htdocs，別名/manual/也可以存取目錄/srv/www/htdocs/manual，(如果DocumentRoot是設定在該目錄，則與別名根本無異)。有了這個別名，http://localhost/manual可以直接存取該目錄。若要使用別名(Alias)指令定義新目標目錄許可權，您可以為它指定目錄(Directory)指令。請參閱節30.6.2, "Directory"。

ScriptAlias

這個指示與Alias類似。此外，它表示目標目錄中的檔案應該當成CGI程序檔來處理。

Server-Side Includes

您可以藉由搜尋所有SSI的可執行檔，啓用Server-Side Includes。按照以下指示，可以完成此目標：

```
<IfModule mod_include.c>
XBitHack on </IfModule>
```

要搜尋SSI的檔案，使用指令`chmod +x` (檔案名稱)，讓檔案可以執行。另一種方法是，明確指定檔案類型來搜尋SSI。按照以下指示，可以完成此目標：

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

建議不要設定.html，因為這樣會造成Apache搜尋SSI的所有網頁(即使確定未包含任何資料的網頁)，而嚴重降低效能。在SUSE LINUX中，這兩個指示已經包括在組態檔，因此正常情況下，不需要變更。

UserDir

藉由`mod_userdir`模組和`UserDir`指令，在使用者的主目錄中指定目錄，可以從主目錄透過Apache來發行檔案。這可以在`SUSEconfig`中設定，方法是設定變數`HTTPD_SEC_PUBLIC_HTML`。要發佈檔案，變數必須設定成`yes`。結果將產生於檔案`/etc/apache2/mod_userdir.conf`中的下列項目，並透過`/etc/apache2/httpd.conf`解譯。

```
<IfModule mod_userdir.c>
UserDir public_html
</IfModule>
```

30.7 使用Apache

要使用Apache顯示靜態網頁，只要將您的檔案放在正確目錄就可以了。在SUSE LINUX中，正確的目錄是`/srv/www/htdocs`。幾張小的範例網頁可能已經安裝在該處。使用這些網頁來檢查Apache是否已正確安裝並正在執行中。接下來，您只需要覆寫或解除安裝這些網頁。自訂CGI程序檔安裝在`/srv/www/cgi-bin`。

作業過程中，Apache會將日誌訊息寫入檔案`/var/log/httpd/access_log`或`/var/log/apache2/access_log`。這些訊息會顯示所要求的資源、傳送的時間以及使用的方法(GET、POST等等)。錯誤訊息記錄至`/var/log/apache2`。

30.8 主動內容

Apache 會提供多種方式來傳遞主動內容。主動內容是根據用戶端輸入的變數資料，而產生的HTML網頁，例如搜尋引擎會回應一或多個輸入的搜尋字串(可能與AND或OR的邏輯運算子交互連結)，傳回包含這些搜尋字串的網頁。

Apache 提供三種產生主動內容的方法：

Server Side Includes (SSI) 這些指示會以特殊註解內嵌在HTML網頁中。Apache 會解譯註解的內容，並將結果當成HTML網頁的一部份來傳遞。

通用閘道介面(CGI) 這些是特定目錄中的程式。Apache 會將用戶端傳輸的參數，轉送至這些程式，然後傳回程式的輸出。此類的程式設計相當簡單，特別是因為現有指令行程式，可以設計成Apache接收輸入，並將輸出傳回Apache。

模組 Apache 提供介面，在要求處理的範圍中執行任何模組。Apache 可以讓這些程式存取重要資訊，例如要求或HTTP標題。程式可以參與主動內容以及其他功能的產生，例如驗證。這些模組的程式設計需要一些專業。此方法的優點是比SSI和CGI，有更佳的效能和可能性。

雖然CGI程序檔是直接由Apache以其擁有者的使用者ID來執行，不過模組是由Apache內嵌的常駐解譯器所控制。這樣一來，不必為每一個要求都啟動和中止一個程序(這會造成程序管理、記憶體管理等等相當大的負荷)。相反的，程序檔是由使用網頁伺服器ID執行的解譯器來處理。

不過，此方法也有其問題。與模組相比，CGI程序檔對不仔細的程式設計較具容忍度。使用CGI程序檔時，無法釋放資源和記憶體之類的錯誤，不會有持續效應，因為在要求處理完畢後，程式即中止。結果是，因程式設計錯誤，而未釋放的記憶體，都予以清除。而使用模組時，因為解譯器屬常駐型，所以程式設計的錯誤效應會一直累積。如果伺服器未重新啟動而且解譯器執行數月，資源無法釋放，例如資料庫連線，會相當麻煩。

30.8.1 Server Side Includes

Server-Side Includes (SSI) 是內嵌在特殊註解中的指示，並由Apache執行。結果內嵌在輸出中。例如，目前的日期可以使用`<!--#echo var="DATE_LOCAL"-->`來列印。左方註解標記(`<!--`)結尾的`#`，會讓Apache知道這是一個SSI指示，不是簡單的註解。

有多種方法可以啓動SSI。最簡單的方法是搜尋SSI的所有可執行檔。另一種方法是指定特定檔案類型來搜尋SSI。二種設定都會在節30.6.2, "Server-Side Includes"說明。

30.8.2 通用閘道介面(CGI)

CGI是*common gateway interface* (通用閘道介面)的縮寫。使用CGI時，伺服器不只傳送靜態HTML，它還會執行一個產生網頁的程式。這使得網頁的產生，是呈現計算的結果，例如搜尋資料庫的結果。將引數傳送給已執行的程式後，程式可以針對每一個要求傳回個別的回應網頁。

CGI的主要優點是這項技術相當簡單。程式只要存在特定目錄中，就能由網頁伺服器執行，和指令行程式一樣。伺服器會在標準輸出通道(stdout)，將程式輸出傳送至用戶端。

理論上，CGI程式可以用任何程式設計語言來編寫。通常，程序檔語言(解譯的語言)，例如Perl或PHP，是用於此用途的。如果速度是重要考量，C或C++可能更合適。

在最簡單的案例中，Apache會在特定目錄(cgi-bin)中尋找這些程式。此目錄可以設定在組態檔，請參閱節30.6, "組態")。需要時，可以指定其他目錄。在此情況下，Apache會在這些目錄中搜尋可執行檔。不過，這樣會有安全風險，因為任何使用者都可以讓Apache執行程式，而有些人可能不懷好意。如果可執行程式被限制在cgi-bin，管理員就可輕易看出誰在此目錄中放了什麼程序檔和程式，然後檢查它們是否有任何不軌。

30.8.3 GET和POST

透過GET或POST，可以輸入參數傳送至伺服器。視使用的方法，伺服器會用不同的方法，將參數傳送至程序檔。使用POST時，伺服器會在標準輸入(stdin)中將參數傳送至程式。主控台啓動時，程式會以相同方法接收它的輸入。使用GET時，伺服器會使用環境變數QUERY_STRING，將參數傳送至程式。

30.8.4 使用模組產生主動內容

許多模組可搭配Apache使用。“模組”這個名詞有兩種不同含意。第一種，有些模組可以與Apache整合，處理特定的功能，例如先前提到的嵌入式程式語言的模組。

第二，與程式設計語言有關，模組是指函數、類別和變數的獨立群組。這些模組是整合在程式中，提供特定功能，例如CGI模組可供所有程式檔語言使用。這些

模組有助於CGI應用程式的程式設計功能，因為它提供各種函數，例如用於讀取參數的方法以及用於HTML輸出的方法。

30.8.5 mod_perl

Perl是一種受歡迎的、廣為使用的程序檔語言。有許多的模組和程式庫供Perl使用，包含一個擴充Apache組態檔案的程式庫。許多Perl適用的程式庫可以從Comprehensive Perl Archive Network (CPAN)取得，網址是<http://www.cpan.org/>。

設定mod_perl

若要在SUSE LINUX設定mod_perl，只要安裝個別的套件即可(請參閱節30.5, "安裝")。安裝之後，Apache組態檔案包含了需要項目(請參閱/etc/apache2/mod_perl-startup.pl)。mod_perl相關資訊請參閱<http://perl.apache.org/>。

mod_perl 與CGI

在最簡單的案例中，將以前的CGI程序檔當成mod_perl程序檔來執行，要使用另一個URL來要求它。組態檔包含別名，它們指向相同的目錄並透過CGI或mod_perl執行它所包含的任何程序檔。所有這些項目已經存在於組態檔中。CGI的別名項目是：

```
ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
```

mod_perl的項目是：

```
<IfModule mod_perl.c>
# Provide two aliases to the same cgi-bin directory,
# to see the effects of the 2 different mod_perl modes.
# for Apache::Registry Mode
ScriptAlias /perl/      "/srv/www/cgi-bin/"
# for Apache::Perlrun Mode
ScriptAlias /cgi-perl/  "/srv/www/cgi-bin/"
</IfModule>
```

以下也是mod_perl需要的項目。這些項目已經存在於組態檔中。

```

#
# If mod_perl is activated, load configuration information
#
<IfModule mod_perl.c>
PerlRequire /usr/include/apache/modules/perl/startup.perl
PerlModule Apache::Registry

#
# set Apache::Registry Mode for /perl Alias
#
<Location /perl>
SetHandler perl-script
PerlHandler Apache::Registry
Options ExecCGI
PerlSendHeader On
</Location>

#
# set Apache::PerlRun Mode for /cgi-perl Alias
#
<Location /cgi-perl>
SetHandler perl-script
PerlHandler Apache::PerlRun
Options ExecCGI
PerlSendHeader On
</Location>

</IfModule>

```

這些項目會為Apache::Registry和Apache::PerlRun模式建立別名。兩種模式的差異如下：

Apache::Registry 所有程序檔都編譯並保留在快取中。每一個程序檔都會套用成副程式的內容。雖然對於效能有好處，不過有一個缺點：程序檔的設計必須非常謹慎，因為變數和副程式會持續存在於要求之間。這表示您必須重設變數，才能讓它們用於下一個要求。例如，如果客戶的信用卡號碼是儲存在線上銀行程序檔的變數中，當下一個客戶使用應用程式並要求相同程序檔時，此號碼可能會再次顯示。

Apache::PerlRun 每一個要求提出時，程序檔都會重新編譯。名稱空間中的變數和副程式，在前後要求之間會消失(名稱空間是所有變數名稱和常式名稱的總稱，它是在程序檔存在期間的指定時間所定義的)。因此，Apache::PerlRun不會有惱人的程式設計問題，因為當程序檔啟動，而且先前的要求並未保留任何值的時候，會重新啓始所有變數。因此，Apache::PerlRun速度比Apache::Registry慢，但是仍然比CGI快很多(雖與CGI有一些相似性)，原因是沒有為解譯器啟動個別程序。

30.8.6 mod_php4

PHP 是一種程式設計語言，專為配合網頁伺服器所開發。PHP 指令是內嵌在 HTML 網頁(類似SSI)中，這和其他語言的指令是儲存在另一個檔(程序檔)很不一樣。PHP 解譯器會處理 PHP 指令，並將處理結果內嵌至 HTML 網頁。

PHP 的首頁是 <http://www.php.net/>。要讓 PHP 可以使用，請為 Apache 2 安裝 `mod_php4-core` 和 `apache2-mod_php4`。

30.8.7 mod_python

Python 是一種物件導向的程式設計語言，語法相當明白和有效。它有一項不普遍，但方便的特色，就是依靠縮排的程式結構。區塊的定義不使用大括弧(例如在 C 和 Perl 中)或其他分界元素，像是 `begin` 和 `end`，而是透過它們的縮排層次。要安裝的套件是 `apache2-mod_python`。

如需這種語言的詳細資訊，請參閱 <http://www.python.org/>。如需關於 `mod_python` 的詳細資訊，請造訪 URL <http://www.modpython.org/>。

30.8.8 mod_ruby

Ruby 是一種相當新的物件導向、高層次的程式設計語言，它很像 Perl 和 Python 的某些方面，適用於程序檔。和 Python 一樣，它有一個清楚、有效的語法。另一方面，Ruby 採用縮寫，例如以 `$.r` 來表示輸入檔中的最後一行數字。有些程式設計師很喜歡這種特點，但也有些人很不喜歡使用。Ruby 的基本概念很像 Smalltalk。

Ruby 的首頁是 <http://www.ruby-lang.org/>。Apache 模組可用於 Ruby。首頁是 <http://www.modruby.net/>。

30.9 虛擬主機

使用虛擬主機，可以在單一網路伺服器裝載多個領域。使用此方法，每個領域不需要都有獨立的伺服器，可以節省成本和管理負荷。虛擬主機有多個選項：

- 以名稱為基礎的虛擬主機
- 以 IP 為基礎的虛擬主機
- Apache 在一台機器上進行多重例項的操作

30.9.1 以名稱爲基礎的虛擬主機

使用以名稱爲基礎的虛擬主機，一個Apache例項可以裝載多個領域。您不需要爲一台機器設定多個IP。這是最簡單、最受歡迎的替代方法。反對使用名稱爲基礎的虛擬主機，其原因請參閱Apache文件。

直接透過設定檔/etc/apache2/httpd.conf進行設定。若要啓用以名稱爲基礎的虛擬主機，請指定合適的指示。NameVirtualHost *. * 可以提示Apache接受所有進來的要求。接下來，設定個別主機：

```
<VirtualHost *>
    ServerName www.example.com
    DocumentRoot /srv/www/htdocs/example.com
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com-error_log
    CustomLog /var/log/apache2/www.example.com-access_log common
</VirtualHost>

<VirtualHost *>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>
```

也必須爲伺服器上原始裝載的領域(www.example.com)設定VirtualHost項目。在這個例子中，原始領域和另一個領域(www.myothercompany.com)裝載在同一部伺服器。

和NameVirtualHost一樣，VirtualHost的指示也會使用*號。Apache使用HTTPS標頭中的主機欄位，將要求連線至虛擬主機。要求會轉送至虛擬主機，該虛擬主機的名稱與此欄位所指定的主機名稱ServerName相符。

至於指示ErrorLog和CustomLog，日誌檔不需要包含領域名稱。在此，使用您選擇的名稱。

ServerAdmin會指定負責人的電子郵件地址，可以在發生問題時洽詢。發生錯誤時，Apache會在傳送給用戶端的錯誤訊息中提供此地址。

30.9.2 以IP爲基礎的虛擬主機

此替代方法需要爲一台機器設定多個IP。如果採用這個方法，一個Apache例項可裝載多個領域，每個領域會被指定一個不同的IP。以下範例顯示如何設定Apache，以裝載原始IP(192.168.1.10)和兩個附加IP(192.168.1.20和192.168.1.21)的兩個附加領域。此特殊範例只在內部網路中有效，因爲從192.168.0.0到192.168.255.0之間的IP，不會在網際網路上傳送。

設定IP 別名

要讓Apache 裝載多個IP，承載的機器必須接受多個IP 的要求。此稱為多重IP 裝載。為達此目的，必須啟動核心中的IP 別名。這是SUSE LINUX 中的預設值。

核心的IP 別名設定好之後，就可以使用指令`ifconfig` 和`route` 來設定主機上的其他IP。這些指令必須以`root` 身份執行。以下範例是假設主機已經有自己的IP，像是`192.168.1.10`，它已被指定給網路設備`eth0`。

輸入指令`ifconfig`，檢視主機IP。使用下列的指令，可以新增其他IP：

```
ip addr add 192.168.1.20/24 dev eth0
```

所有IP 都已指定給相同的實體網路設備(`eth0`)。

多個IP 的虛擬主機

在系統上設定好IP 別名，或者主機已經設定多個網路卡時，就可以設定Apache。請為每一個虛擬伺服器指定個別的`VirtualHost` 區域：

```
<VirtualHost 192.168.1.20>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>

<VirtualHost 192.168.1.21>
    ServerName www.anothercompany.com
    DocumentRoot /srv/www/htdocs/anothercompany.com
    ServerAdmin webmaster@anothercompany.com
    ErrorLog /var/log/apache2/www.anothercompany.com-error_log
    CustomLog /var/log/apache2/www.anothercompany.com-access_log common
</VirtualHost>
```

`VirtualHost` 指示只會指定給附加領域。原始領域(`www.example.com`) 是由它自己的設定值來設定(在`DocumentRoot` 下方等等)，位於`VirtualHost` 區塊之外。

30.9.3 多個Apache 例項

領域管理員透過上述提供虛擬主機的方法，可以讀取其他領域的資料。若要區分個別領域，請啟動數個Apache 例項，每個例項在組態檔都設定自己的`User`、`Group` 以及其他指示。

在組態檔中，使用監聽(`Listen`) 指示，指定個別Apache 例項所負責的IP。就以上的範例而言，第一個Apache 例項的指示如下：

```
Listen 192.168.1.10:80
```

其它兩個例項的指示如下：

```
Listen 192.168.1.20:80
```

```
Listen 192.168.1.21:80
```

30.10 安全性

30.10.1 將風險降到最小

如果您不需要機器上的網頁伺服器，請在runlevel 編輯器中將Apache 停用、解除安裝，或者一開始就不要安裝。若要將風險降到最小，請關閉所有不需要的伺服器。這對當成防火牆的主機特別適用。可能的話，不要在這些主機執行任何伺服器。

30.10.2 存取許可權

DocumentRoot 應該屬於root

根據預設，DocumentRoot 目錄(/srv/www/htdocs) 和CGI 目錄屬於root 使用者。您不應該變更此設定。如果所有人都可以寫入目錄，任何使用者都可以將檔案放到目錄中。接著，這些檔案就可以讓使用wwwrun 使用者許可權的Apache 來執行。另外，Apache 對於所傳送的資料或程序檔，不應該擁有任何寫入許可權。因此，這些不應該屬於wwwrun 使用者，而屬於其他使用者，例如root。

要讓使用者可以將檔案放到Apache 的文件目錄，不要讓所有人都可以寫入。應該要做的是，建立一個大家都可寫入的子目錄，例如/srv/www/htdocs/miscellaneous。

從主目錄發佈文件

如果要允許使用者發佈檔案，可以將其主目錄下的一個子目錄宣告為網路發佈用途。習慣上，這個子目錄叫做~/public_html。在SUSE LINUX 中，預設為啓動此功能。如需詳細資料，請參閱節30.6.2, "UserDir"。

在URL 中指定使用者，就可以存取這些網頁。URL 包含的元素~使用者名稱，是做為使用者主目錄中各個目錄的捷徑。例如，在瀏覽器輸入http://localhost/~tux，即可列示使用者tux 主目錄中，public_html 目錄的檔案。

30.10.3 保持更新

如果您運作一個網頁伺服器，尤其當這個網頁伺服器是公開存取時，要隨時知道設計問題以及潛在的漏洞。漏洞和修正程式的來源列示在節30.12.3, “安全性”。

30.11 疑難排解

如果發生問題，例如Apache未顯示網頁或者未正確顯示網頁，以下方法可以協助您找到問題所在。首先，請查閱錯誤日誌，檢查它所包含的訊息是否顯示出錯誤。一般錯誤日誌檔案是/var/log/apache2/error_log。

可靠的方法是追蹤主控台的日誌檔，了解伺服器對存取的反應。操作方法是，在root主控台輸入以下指令。

```
tail -f /var/log/apache2/*_log
```

查閱<http://bugs.apache.org/> 的線上問題資料庫。閱讀相關郵件清單和新聞群組。使用者的郵件清單可以從<http://httpd.apache.org/userslist.html> 取得。推薦的新聞群組是comp.infosystems.www.servers.unix 以及相關群組。

如果上述管道都無法提供解決方法，而且您確定Apache發生問題，請到<http://www.suse.de/feedback/> 報告問題。

30.12 若需詳細資訊

Apache是廣泛使用的網頁伺服器。所以，有許多延伸文件以及許多協助和支援網站。

30.12.1 Apache

Apache同時提供詳細的文件。文件中的安裝部份，請參閱節30.5, “安裝”。安裝之後，可在<http://localhost/manual> 取得文件。可以從<http://httpd.apache.org> 的Apache首頁取得最新的文件。

30.12.2 CGI

以下網頁提供CGI的詳細資訊：

- <http://apache.perl.org/>
- <http://perl.apache.org/>
- <http://www.modperl.com/>
- <http://www.modperlcookbook.org/>
- <http://www.fastcgi.com/>
- <http://www.boutell.com/cgic/>

30.12.3 安全性

SUSE LINUX 套件的最新修補程式可以從<http://www.novell.com/linux/security/securitysupport.html> 取得。定期造訪此URL。在此，您可以註冊加入SUSE 郵件清單，即可收到安全性通知。

Apache 小組正在推廣一種關於Apache 設計問題的開放資訊原則。最新的設計問題報告以及可能的漏洞會發佈於http://httpd.apache.org/security_report.html。如果您偵測到安全性設計問題(檢查前述網頁，確定尚未發現該設計問題)，請向security@suse.de 或security@apache.org 報告問題。

Apache 的其他安全性問題相關資訊的資源(以及其他網際網路程式)：

- <http://www.cert.org/>
- <http://www.vnunet.com/>
- <http://www.securityfocus.com/>

30.12.4 其他來源

如果您遇到困難，請查閱「SUSE 支援資料庫」，網址是<http://portal.suse.com/sdb/en/index.html>。以Apache 為主題的線上報紙，網址是<http://www.apacheweek.com/>。

Apache 的歷程網址是http://httpd.apache.org/ABOUT_APACHE.html。此頁面也說明稱伺服器為Apache 的原因。

關於從1.3 升級至2.0 版的資訊，可由下列網址取得：<http://httpd.apache.org/docs-2.0/en/upgrading.html>。

檔案同步化

今日有許多使用數台電腦—在家使用一台、在辦公室使用一台或數台電腦，而且可能在外邊還使用膝上型電腦或PDA。許多檔案都需要各存一份在所有這些電腦上。您可能希望能在每一部電腦上工作、修改檔案，之後還能讓所有的電腦都擁有最新的資料。

31.1	可用的資料同步化軟體	454
31.2	選取程式時所要考慮的決定性因素	455
31.3	Unison 的介紹	458
31.4	CVS 簡介	460
31.5	Subversion 簡介	462
31.6	rsync 簡介	465
31.7	mailsync 簡介	467

31.1 可用的資料同步化軟體

對於永久以快速網路連結的電腦而言，資料同步化不是問題。在此狀況下，使用NFS之類的網路檔案系統，並將檔案儲存在伺服器上，所有主機就可以透過網路來存取相同的資料。但如果網路連線品質很差或不是永久的，就無法使用此方法。當您出門在外使用膝上型電腦時，所有需要的檔案複本都必須在本機硬碟上。不過之後需要將修改過的檔案同步化。當您在某一台電腦上修改檔案時，請確定也更新了所有其他電腦上的同一檔案。至於一些零散的複本，則可以使用scp或rsync手動更新它。然而，如果有許多檔案需要處理，則該程序有可能變得很複雜，而且需要更小心才能避免類似以舊檔案覆寫新檔案的錯誤。

┌

資料遺失的風險

在您開始使用同步化系統管理資料之前，應該先好好的認識要使用的程式並測試其功能。對於重要檔案而言，備份是不可或缺的動作。

└

手動同步化資料非常耗時，而且是一種容易發生錯誤的工作，要避免這些缺點，可以使用一種以多種方法將此工作自動化的程式。下列摘要讓您概略了解這些程式的運作方式和使用方法。如果您打算使用它們，請詳閱程式文件。

31.1.1 Unison

Unison不是網路檔案系統。實際上，只是在本機儲存和編輯檔案。您可以手動執行Unison程式，將檔案同步化。第一次執行同步化時，會在兩個主機上建立資料庫，包含總和檢查碼、時戳以及選取檔案的權限。下次執行時，Unison可以識別哪些檔案已變更，並建議要從其他主機傳輸或傳輸至其他主機。通常可以接受所有的建議。

31.1.2 CVS

CVS主要是用來管理程式來源版本，它使得檔案複本可以保留在多台電腦上。因此，它也適用於資料同步化。CVS負責維護伺服器上的中央儲存區，檔案及檔案的變更都儲存在此。在本機上執行的變更會交付至儲存區，並且可以利用更新從其他電腦擷取。兩個程序都必須由使用者起始。

CVS對於錯誤有非常大的彈性與包容性，所以可以應付多台電腦同時進行變更的情況。變更會合併，而且如果在相同行發生變更，就會報告衝突。衝突發生時，資料庫會維持一致的狀態。只有在用戶端主機上才能看到衝突，並加以解決。

31.1.3 subversion

subversion 與不斷演變的CVS大不相同，subversion 是一脈相承的開發計劃。subversion 是CSV 的技術改善後繼加強版。

與舊一代的產品相比，subversion 在許多方面都已大幅改善。由於其傳統包袱之故，CVS 只能維護檔案但卻無法維護目錄。目錄在subversion 中也有版本歷史，而且可以像檔案一樣進行複製和重新命名。它也可以將中繼資料新增至每個檔案以及每個目錄。使用版本設定就可以充份維護此中繼資料。與CVS 相反的是，subversion 支援透過專用通訊協定進行通透網路存取，例如WebDAV (以Web 為基礎的分散式製作與版本設定)。WebDAV 擴充了HTTP 通訊協定的功能，而能對遠端網頁伺服器上的檔案進行協同寫入存取。

subversion 大部份是由已經存在的軟體套件組合而成。因此，Apache 網頁伺服器與WebDAV 延伸程式總是與subversion 一起執行。

31.1.4 mailsync

mailsync 與前面小節所述的同步化工具不同，它只能同步化信箱之間的電子郵件。這個程序可以套用至本機信箱檔案以及IMAP 伺服器上的信箱。

根據電子郵件標頭所包含的訊息ID，將個別訊息同步化或刪除。在個別信箱之間與信箱階層之間都可以同步化。

31.1.5 rsync

當不需要版本控制，但是大型目錄結構需要透過緩慢的網路連線進行同步化時，rsync 工具針對僅傳送檔案中的變更可提供已開發成熟的機制。這不只包含文字檔，也包含二進位檔。為了偵測檔案之間的不同，rsync 會將檔案分為區塊並計算它們的檢查總數。

偵測變更將需要付出相當大的成本。要同步化的系統應該要具有相當的硬體配備，才能使用rsync。RAM 尤其重要。

31.2 選取程式時所要考慮的決定性因素

31.2.1 用戶端對伺服器與點對點

配送資料時常使用的模式有兩種。第一個模式是，所有的用戶端都以中央伺服器為準，將其檔案同步化。伺服器至少必須偶爾可以讓所有的用戶端存取。subversion、CVS 以及WebDAV 都採用這種模式。

另一種可能性就是，讓網路上所有主機都以點對點的方式將彼此間的資料同步化。這是unison 所遵循的概念，rsync 實際上使用的是用戶端模式，但是任何用戶端都可以當成伺服器使用。

31.2.2 可攜式

在許多其他的作業系統上也可以使用subversion、CVS 以及unison，包含各種Unix 與Windows 系統。

31.2.3 互動式與自動化

在subversion、CVS、WebDAV 以及Unison 中，資料同步化是由使用者以手動方式啓動。這讓使用者對於要同步化的資料進行良好的控制，並可輕鬆地處理衝突。然而，如果同步化間隔太長，就比較可能發生衝突。

31.2.4 衝突：事件與解決方案

即使有數個人員同時在某個大型的程式專案上一起工作，在subversion 或CVS 中發生衝突的機率還是相當地少。這是因為文件是在個別的行列上進行合併。當發生衝突時，只有一個用戶端會受到影響。subversion 或CVS 中的衝突通常都可以輕易解決。

Unison 會報告衝突，並允許將受影響的檔案排除在同步化之外。然而，變更的合併不像在subversion 或CVS 中那樣容易。

兩相比較下，subversion 或CVS 在發生衝突時可以部份接受變更，但WebDAV 只會在成功地進行完整修改時執行登記。

rsync 中則無衝突處理功能。使用者必須小心不要覆寫檔案，並手動解決所有可能的衝突。基於安全著想，可以另外使用RCS 這一類的版本設定系統。

31.2.5 選取和新增檔案

在其標準組態中，Unison 會同步化整個目錄樹狀結構。出現在樹狀結構中的新檔案會自動包括在同步化中。

在subversion 或CVS 中，必須分別使用svn add 或cvs add 指令，以明確地新增目錄與檔案。這讓使用者對於要同步化的檔案擁有更大的控制權。另一方面，新檔案時常會被忽略，特別是在處理大量的檔案而忽略了svn update 以及svn status 或cvs update 輸出中的問號時。

31.2.6 歷程記錄

subversion 或CVS 還有另一項功能，就是可以重新建構舊的檔案版本。每個變更都可以插入簡短的編輯符號，而且之後可以根據其內容與符號輕易地追蹤檔案的發展。這對論文與程式文字而言，是一種很珍貴的助力。

31.2.7 資料量與硬碟需求

所有相關主機的硬碟都需要有足夠的可用空間來儲存所有分散式的資料。subversion 與CVS 在伺服器上還需要額外的空間，供儲存區資料庫使用。檔案歷程記錄也會儲存在伺服器上，因此需要更多的空間。當文字格式的檔案變更時，只需儲存修改過的那幾行。每當變更檔案時，二進位檔案就會需要與該檔案大小相同的額外空間。

31.2.8 GUI

Unison 提供圖形使用者介面，可顯示Unison 所要執行的同步化程序。接受提議或排除同步化中的個別檔案。在文字模式中，以互動式的方式確認個別程序。

有經驗的使用者通常會從指令行執行subversion 或CVS。然而，在Linux 系統中也有cervisia 之類的圖形使用者介面，其他作業系統中也有wincvs 之類的圖形使用者介面。許多開發工具(像是kdevelop) 以及文字編輯器(像是emacs) 都支援CVS 或subversion。使用這些前端程式的話，衝突的解決方案通常會更容易執行。

31.2.9 使用者親切性

Unison 與rsync 使用簡便，而且也適於新手使用。CVS 與subversion 則比較難操作。使用者必須瞭解儲存區與本機資料之間的互動。對資料的變更應該先在本機上與儲存區合併。這是使用cvs update 或svn update 指令來執行。接著必須使用cvs commit 或svn commit 指令將資料傳送回儲存區。只要瞭解此程序，即使是新手也能輕鬆使用CVS 或subversion。

31.2.10 防止攻擊的安全性

在傳輸期間，應該保護資料以防攔截和竄改。Unison、CVS、rsync 以及subversion 都可以透過ssh (安全的外圍程式) 來使用，提供安全性以防護此類攻擊。應該避免透過rsh (遠端外圍程式) 執行CVS 或Unison。同樣地不建議在不安全的網路中使用pserver 機制存取CVS。subversion 與Apache 一起執行，便提供了必要的安全性措施。

31.2.11 針對資料遺失的防護

開發人員使用CVS來管理程式專案已經有很長的一段時間，而且極為穩定。因為開發的歷程記錄皆已儲存，所以CVS甚至提供保護，防止某些使用者錯誤發生，例如不小心刪除檔案。儘管subversion不像CVS一樣普及，它已運用於像是subversion專案本身的生產環境中。

Unison仍算是相當新的產品，但是宣稱擁有高水準的穩定性。然而，它對於使用者錯誤更為敏感。一旦確認要將檔案刪除同步化，就不能再還原檔案。

Table 31.1: 檔案同步化工具的功能：-- = 非常差、- = 差或無法使用、o = 中等、+ = 好、++ = 佳、x = 可用

	unison	CVS/subv.	rsync	mailsync
用戶端/伺服器	相等	C-S/C-S	C-S	相等
可攜式	Lin、Un*x、Win	Lin、Un*x、Win	Lin、Un*x、Win	Lin、Un*x
互動	x	x/x	x	-
速度	-	o/+	+	+
衝突	o	++/++	o	+
檔案Sel。	目錄	選項/檔案、目錄	目錄	信箱
歷程記錄	-	x/x	-	-
硬碟空間	o	--	o	+
GUI	+	o/o	-	-
困難度	+	o/o	+	o
攻擊	+(ssh)	+/(ssh)	+(ssh)	+(SSL)
資料損失	+	++/++	+	+

31.3 Unison 的介紹

Unison是同步化和傳輸整個目錄樹狀結構的最佳解決方案。同步化是以雙向執行，而且可以透過直覺式圖形前端程式來控制。也可以使用主控台版本。同步化也可以自動化，因此不需要與使用者互動，但是經驗是必要的。

31.3.1 需求

Unison 必須安裝在用戶端以及伺服器上。在此環境中，「伺服器」一詞是指第二部遠端主機(與CVS不同，詳述於節31.1.2, “CVS”)。

在下一小節中，Unison 是與ssh一起使用。在此例中，必須在用戶端上安裝SSH用戶端，並在伺服器上安裝SSH伺服器。

31.3.2 使用Unison

Unison 所使用的方法是兩個目錄(*roots*)彼此互相關聯。此關聯是象徵性的—它不是線上的連線。以此為例，目錄配置如下列所示：

```
_____  
用戶端： /home/tux/dir1  
伺服器： /home/geeko/dir2  
_____
```

您想要同步化這兩個目錄。使用者在用戶端上的身份為tux，在伺服器上的身份為geeko。第一件要做的事就是測試用戶端與伺服器之間的通訊是否可運作：

```
unison -testserver /home/tux/dir1 ssh://geeko@server//homes/geeko/dir2
```

最常遭遇的問題為：

- 在用戶端與伺服器上使用的Unison版本是不相容的。
- 伺服器並不允許SSH連線。
- 兩個指定的路徑都不存在。

如果每個項目都可以運作，請省略-testserver選項。在第一次同步化期間，Unison並不知道兩個目錄之間的關係，並且會針對個別檔案與目錄的傳輸方向送出建議。在‘動作’資料欄中的箭頭會指出傳輸的方向。問號表示Unison無法建議傳輸方向，因為兩個版本都已經變更或者是新的版本。

方向鍵可以用來設定個別項目的傳輸方向。如果所有顯示項目的傳輸方向都正確，則只要按一下‘到’。

若要控制Unison的特性(例如，對於沒有衝突錯誤的檔案是否要自動執行同步化)，可以在啟動該程式時，利用指定的指令行參數來控制。使用unison --help可檢視所有參數的完整清單。

Example 31.1: `~/unison/example.prefs` 檔案

```
root=/home/tux/dir1
root=ssh://wilber@server//homes/wilber/dir2
batch=true
```

使用者目錄`~/unison`會為每一組同步化資料保留一個同步化日誌。組態集合，像是`~/unison/example.prefs`，也可以儲存在此目錄中。若要啟動同步化，請依照`unison example.prefs`中所示，將此檔案指定為指令行參數。

31.3.3 若需詳細資訊

Unison 的官方文件特別有用。因此，本小節只提供簡短的介紹。完整的手冊可在<http://www.cis.upenn.edu/~bcpierce/unison/>中以及在SUSE套件`unison`中找到。

31.4 CVS 簡介

如果經常編輯某些個別檔案，並且以檔案格式儲存，例如ASCII文字或程式來源文字，就非常適合使用CVS進行同步化。您可以使用CVS同步化其他格式的資料，例如JPEG檔案，但是會造成大量的資料，因為檔案的所有變體都會永久儲存在CVS伺服器上。在這種情形下，CVS的大部份功能都將無法使用。只有在所有工作站都可以存取同一伺服器時，才能使用CVS同步化檔案。

31.4.1 設定CVS伺服器

*server*是所有有效檔案所在的主機，這包含所有檔案的最新版本。任何靜態工作站都可做為伺服器。可能的話，CVS儲存區的資料應該包含在定期備份中。

當設定CVS伺服器時，透過SSH授與使用者該伺服器的存取權可能是不錯的方式。如果使用者是以`tux`的身份登入伺服器，而且CVS軟體既安裝在伺服器上也安裝在用戶端上，則必須在用戶端上設定下列環境變數：

```
CVS_RSH=ssh CVS_ROOT=tux@server:/serverdir
```

`cvsexport` 指令可用來從用戶端啓始CVS伺服器。這只需要做一次。

最後，必須為同步化指定名稱。在用戶端上選取或建立一個目錄，專門存放CVS所管理的檔案(目錄也可以是空的)。該目錄的名稱也就是同步化的名稱。在此範例中，該目錄稱為`synchome`。變更至此目錄，然後輸入下列指令將同步化名稱設為`synchome`：

```
cvsexport synchome tux wilber
```

許多CVS指令都需要註解。因此，CVS會啓動編輯器(如果没有定義編輯器，則會啓動環境變數`$EDITOR`或`vi`中所定義的編輯器。)在指令行先輸入註解，就可以避免編輯器的呼叫，如下列範例所示：

```
cvsexport -m 'this is a test' synchome tux wilber
```

31.4.2 使用CVS

現在可以使用`cvsexport synchome`從所有主機取出同步化儲存區。這會在用戶端上建立新的子目錄`synchome`。若要將變更交付至伺服器，請換至`synchome`目錄(或其中一個子目錄)，然後輸入`cvsexport`。

根據預設，所有的檔案(包括子目錄)都會交付給伺服器。若只要交付個別的檔案或目錄，請在`cvsexport file1 directory1`中指定它們。在將它們交付至伺服器前，必須使用像是`cvsexport file1 directory1`之類的指令，將新檔案與目錄新增至儲存區。接著，使用`cvsexport file1 directory1`交付新增的檔案與目錄。

您更換至另一個工作站時，如果先前的工作階段尚未取出該工作站的同步化儲存區，請現在取出(參閱上方)。

使用`cvsexport`啓動與伺服器的同步化。依

照`cvsexport file1 directory1`中的個別檔案或目錄來更新。

若要查看目前檔案與伺服器上所存版本的差異，請使用`cvsexport diff`

或`cvsexport diff file1 directory1`指令。使用`cvsexport -nq export`來查看哪些檔案會受到更新的影響。

以下是更新期間所顯示的一些狀態符號：

- U 已更新本機版本。這將會影響伺服器所提供及在本地系統上所遺失的全部檔案。
- M 已修改本機版本。如果在伺服器上有一些變更，可以在本機複本中將差異合併。

- P 本機版本已使用伺服器版本修補。
- C 本機檔案與儲存區中目前的版本衝突。
- ? 這個檔案並不存在於CVS。

M 狀態是指在本機修改過的檔案。或是將本機複本交付至伺服器，或是移除本機檔案後再執行一次更新。在此例中，會從伺服器擷取遺失的檔案。如果您交付本機已修改的檔案，而該檔案是在同一行中變更並交付，則有可能造成衝突，這是以C表示。

如果發生這種狀況，請查看檔案中的衝突記號(*;* 與*j*)，並在兩個版本之間做一選擇。這有可能是相當麻煩的工作，您可能會決定捨棄變更、刪除本機檔案，然後輸入 `cvs up`，以便從伺服器擷取目前的版本。

31.4.3 若需詳細資訊

本小節僅針對CVS的許多可能性提供簡短的介紹。下列URL提供許多相關文件：

<http://www.cvshome.org/>
<http://www.gnu.org/manual/>

31.5 Subversion 簡介

Subversion 是免費的開放式來源版本設定控制系統，而且普遍視之為CVS的後繼加強版，意即在CVS中已推出的功能，通常在subversion中也有。另外特別推薦的原因是，它兼具了CVS的優點，卻不必忍受其缺點。許多功能已經在節31.1.3, "subversion" 中都已簡要說明過。

31.5.1 安裝Subversion 伺服器

在伺服器上安裝儲存區資料庫是單單簡單的程序。Subversion 為此提供了專用的管理工具。建立新儲存區所要輸入的指令是：

```
svnadmin create /path/to/repository
```


其他的選項使用 `svnadmin help` 就可列出。與CVS大不相同的是，subversion使用的不是RCS為基礎，而是Berkeley Database。請確定不要在遠端檔案系統上安裝儲存區，例如NFS、AFS或Windows SMB。該資料庫需要POSIX鎖定機制，但是這些檔案系統不支援。

`svnlook` 指令會提供現有儲存區的資訊。

```
svnlook info /path/to/repository
```

伺服器必須經過設定才能讓不同的使用者存取儲存區。您可以使用Apache網頁伺服器以及WebDAV來執行此動作，或使用`svnserve`以subversion來封裝伺服器。只要一啟動和執行`svnserve`，就可以使用`svn://`或`svn+ssh://`URL來存取儲存區。呼叫`svn`時應該要自我驗證的使用者，可以設定在`/etc/svnserve.conf`。

要選用Apache或`svnserve`的決定取決於許多因素。建議您瀏覽subversion的書籍。在節31.5.3, “若需詳細資訊”中可以找到更多它的詳細資訊。

31.5.2 使用與作業

使用`svn`指令(與 `cvs`相似)來存取subversion儲存區。符合對應儲存區且由設定正確之伺服器所提供的內容，可供任何用戶端以下列指令之一存取：

```
svn list http://svn.example.com/path/to/project
```

或

```
svn list svn://svn.example.com/path/to/project
```

使用`svn checkout`指令，將現有的專案儲存在目前的目錄中(取出)：

```
svn checkout http://svn.example.com/path/to/project nameofproject
```

「取出」會在用戶端上建立新的子目錄`nameofproject`。然後就可以執行各項作業(新增、複製、重新命名、刪除)：

```
svn add file
svn copy oldfile newfile
svn move oldfile newfile
svn delete file
```

這些指令也可以用在目錄上。subversion 可以另外記錄檔案或目錄的屬性：

```
svn propset license GPL foo.txt
```

先前的範例會為license 屬性設定GPL 值。使用svnproplist 顯示屬性：

```
svn proplist --verbose foo.txt Properties on 'foo.txt':license :GPL
```

使用svn commit 將變更儲存至伺服器。另一個使用者可以使用svn update 與伺服器同步化，在其工作目錄中併入您的變更。

與CVS 不同的是，subversion 不需要使用svn status 存取儲存區，就可以顯示它的工作目錄狀態。本機變更會顯示在五個資料欄中，而第一個資料欄是最重要的：

- " 沒有變更。
- 'A' 物件已標示為新增。
- 'D' 物件已標示為刪除。
- 'M' 物件已修改。
- 'C' 物件發生衝突。
- 'I' 物件被忽略。
- ?' 物件不是由版本設定控制所維護。
- !' 物件已報告為遺失。當未使用svn 指令而刪除或移動物件時，就會出現旗標。
- ~' 物件是被當做檔案來維護，但是物件曾被目錄所取代，或是目錄被物件所取代。

第二個資料欄會顯示屬性的狀態。在subversion 書籍中可以參閱所有其他欄的意義。

使用指令svn help 以取得某指令的參數描述：

```
svn help proplist
proplist (plist, pl):列出檔案、目錄或版本的所有屬性。
用法：1. proplist [PATH...]
      2. proplist --revprop -r REV [URL]
```

1. 列出工作複本中的版本屬性。
2. 列出儲存區版本上的尚未設定版本之遠端屬性。 ...

31.5.3 若需詳細資訊

最主要的參考資料為<http://subversion.tigris.org/> 的subversion專案首頁。可以在您安裝套件subversion-doc後，於file:///usr/share/doc/packages/subversion/html/book.html目錄中找到推薦的書籍，您也可以在线上找到，網址為<http://svnbook.red-bean.com/svnbook/index.html>。

31.6 rsync 簡介

當有大量的資料需要定期傳輸，但是並無太多變更時，rsync就非常有用。例如，當建立備份時就非常適用。另一個應用程式則將焦點放在建置伺服器。這些伺服器儲存了網頁伺服器的完整目錄樹狀結構，會定期鏡像處理至DMZ中的網頁伺服器。

31.6.1 組態與作業

rsync可以使用兩個模式來操作。它是用來歸檔或複製資料。若要完成此動作，在目標系統上只需要像是ssh的遠端外圍程序。然而，rsync也可以像daemon一樣，用來為網路提供目錄。

rsync的基本作業模式並不需要任何特殊的組態。rsync直接允許將完整的目錄鏡像處理至另一個系統。例如，下列指令可在名為sun的備份伺服器上建立tux主目錄的備份：

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

下列指令可用來還原目錄：

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

到此為止，其處理方式與一般的複製工具(例如scp)的處理沒有太大的差別。

rsync應該在“rsync”模式中操作，才能充分使用其所有的功能。執行方法是在其中一個系統上啟動rsyncd精靈。在/etc/rsyncd.conf檔案中設定它。例如，如果要讓/srv/ftp目錄可供rsync使用，請使用下列組態：

```

gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log

[FTP]
    path = /srv/ftp
    comment = An Example

```

接著使用 `rcrsyncd start` 來啟動 `rsyncd`。在開機程序期間也可以自動啟動 `rsyncd`。設定的方式有兩種，一是在 YaST 所提供的 `runlevel` 編輯器中啟用此服務，另一是手動輸入 `insserv rsyncd` 指令。`rsyncd` 也可以由 `xinetd` 啟動。然而，只有對很少使用 `rsyncd` 的伺服器才建議這麼做。

本範例也建立了傾聽所有連線的日誌檔。這個檔案是儲存在 `/var/log/rsyncd.log`。

這樣才可以從用戶端系統測試傳輸。使用下列指令來執行此動作：

```
rsync -avz sun::FTP
```

此指令會列出伺服器上 `/srv/ftp` 目錄中所有存在的檔案。這個要求也會記錄在 `/var/log/rsyncd.log` 日誌檔中。若要啟動實際的傳輸，請提供目標目錄。使用 `.` 來代表目前的目錄。例如：

```
rsync -avz sun::FTP .
```

依照預設，在使用 `rsync` 同步化時不會刪除檔案。如果要強制刪除檔案，就必須加上額外的選項 `--delete`。若要確保不會刪除較新的檔案，則可以改用 `--update` 選項。任何產生的衝突都必須手動解決。

31.6.2 若需詳細資訊

關於 `rsync` 的重要資訊，請參閱 `man` 頁面。指令為 `man rsync` 和 `man rsyncd.conf`。如需關於 `rsync` 作業原則的技術參考資料，可在 `/usr/share/doc/packages/rsync/tech_report.ps` 中找到。您可以在專案網站 <http://rsync.samba.org/> 上找到關於 `rsync` 的最新消息。

31.7 mailsync 簡介

mailsync 主要適用於下列三項工作：

- 將本機儲存的電子郵件以及儲存在伺服器上的郵件同步化。
- 將信箱移轉至另一種格式或另一個伺服器
- 信箱的完整性檢查或搜尋重複資料

31.7.1 組態與使用

mailsync 可以區分信箱本身(*store*) 以及兩個信箱之間的連線(*channel*)。 「儲存區」與「通道」的定義是儲存在`~/.mailsync` 中。下列段落說明一些儲存區的範例。

簡單的定義可能會如下所示：

```
store saved-messages {
    pat Mail/saved-messages
prefix Mail/
}
```

Mail/ 是使用者主目錄下之子目錄，它包含電子郵件資料夾，其中包括saved-messages 資料夾。如果mailsync 是以`mailsync -m saved-messages` 啓動，它會列出saved-messages 中所有訊息的索引。如果建立了下列定義

```
store localdir {
pat Mail/*
prefix Mail/
}
```

`mailsync -m localdir` 指令會列出所有儲存在Mail/ 之下的訊息。相對的，`mailsync localdir` 指令會列出資料夾名稱。IMAP 伺服器上某儲存區的設定將如下所示：

```
store imapinbox {
server {mail.edu.harvard.com/user=gulliver}
ref {mail.edu.harvard.com}
pat INBOX
}
```

上述範例只能處理IMAP 伺服器的主要資料夾。子資料夾的儲存區將如下所示：

```
store imapdir {  
server {mail.edu.harvard.com/user=gulliver}  
ref {mail.edu.harvard.com}  
pat INBOX.*  
prefix INBOX.  
}
```

如果IMAP 伺服器支援加密的連線，則伺服器設定應該變更為

```
server {mail.edu.harvard.com/ssl/user=gulliver}
```

或者，如果不知道伺服器的憑證，則變更為

```
server {mail.edu.harvard.com/ssl/novalidate-cert/user=gulliver}
```

稍後會說明該字首。

目前在Mail/ 下的資料夾應該連線至IMAP 伺服器的子目錄：

```
channel folder localdir imapdir { msinfo .mailsync.info }
```

mailsync 會使用msinfo 檔案來追蹤已同步化的訊息。

mailsync folder 指令會執行下列動作：

- 展開兩邊的信箱圖樣。
- 從產生的資料夾名稱移除字首。
- 同步化成對的資料夾(如果它們不存在就建立它們)。

因此在IMAP 伺服器上的INBOX.sent-mail 資料夾會與Mail/sent-mail 本機資料夾同步化(如果有前述的定義)。在個別資料夾之間的同步化將會依照下列方式執行：

- 如果在兩邊都已經有某一訊息，則不會執行任何動作。
- 如果在某一邊缺少某訊息而且為新訊息(未列在msinfo 檔案)，就會傳輸至該處。
- 如果訊息僅存在於某一邊而且為舊的(已列於msinfo 檔案)，就會刪除該處的訊息(因為另一邊原先所存在的該訊息已遭刪除)。

若想事先知道同步化期間要傳送哪些訊息，以及要刪除哪些訊息，請使用 `mailsync folder localdir` 來啟動具有通道與儲存區的 `mailsync`。此指令會產生所有在本機主機上列為新訊息的清單，以及同步化期間在 IMAP 那一邊將刪除的所有訊息清單。同樣地，`mailsync folder imapdir` 指令會產生所有在 IMAP 那一邊為新訊息的清單，以及同步化期間在本機主機上將刪除的所有訊息清單。

31.7.2 可能的問題

在資料遺失的事件中，最安全的方法就是刪除相關的通道日誌檔 `msinfo`。因此，僅存在於某一邊的所有訊息會被視為新訊息，所以會在下一次同步化期間予以傳輸。

只有具有訊息 ID 的訊息會包括在同步化中。缺乏訊息 ID 的訊息則會被忽略，亦即不會傳輸或刪除它們。缺乏訊息 ID 通常是有問題的程式在傳送或撰寫訊息時所造成。

在某些 IMAP 伺服器上，會使用 `INBOX` 來處理主要資料夾，而子資料夾則會以隨機選取的名稱來處理（與 `INBOX` 及 `INBOX.name` 相對）。因此，像這樣的 IMAP 伺服器無法為子資料夾指定專用的圖樣。

成功地將訊息傳輸至 IMAP 伺服器後，`mailsync` 所使用的信箱驅動程式 (`c-client`) 會設定特殊的狀態旗標。因此，`mutt` 之類的電子郵件程式就無法將這些訊息識別成新訊息。請使用 `-n` 的選項以停用特殊狀態旗標的設定值。

31.7.3 若需詳細資訊

在 `/usr/share/doc/packages/mailsync/` 中的 `README`，是包括在 `mailsync` 中，它提供更多資訊。在此連線中，RFC 2076「一般網際網路訊息標頭」特別有趣。

Samba

使用Samba，就可以將Unix 機器設為DOS、Windows 以及OS/2 機器的檔案與列印伺服器。本章節介紹Samba 組態的基本概念，並說明可在網路中設定Samba 的YaST 模組。

32.1	設定伺服器	473
32.2	做為登入伺服器的Samba	476
32.3	使用YaST 設定Samba 伺服器	478
32.4	設定用戶端	479
32.5	最佳化	480

Samba 已經是一個開發至完全成熟且相當複雜的產品。本章節提供其基本功能的概觀。在所附的數位文件中有詳細的討論。如果已安裝Samba，若要取得更多的線上文件與範例，請在指令列輸入`apropos samba`以顯示部分手冊頁面或直接瀏覽`/usr/share/doc/packages/samba`目錄。在`examples`子目錄中可以找到加備註的範例組態。`(smb.conf.SuSE)`。

隨附samba 套件版本3 的一些重要新功能：

- 支援Active Directory
- 加強Unicode 支援
- 已全面修訂內部驗證機制
- 改善Windows 200x 與XP 列印系統的支援
- 在Active Directory 領域中可以將伺服器設為成員伺服器
- 採用NT4 領域，允許將NT4 領域移轉至Samba 領域

移轉至Samba3

將Samba 2.x 移至Samba 3 時，有一些特殊的項目需要考慮。此主題的討論包含在「Samba HOWTO 文件集」中，裡面有整個章節專門討論它。在安裝`samba-doc` 套件後，可以在`/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` 中找到HOWTO。

Samba 使用依據NetBIOS 服務的SMB 通訊協定(伺服器訊息區塊)。由於來自IBM、Microsoft 發行了此通訊協定的壓力，因此其他的軟體製造商可以建立對Microsoft 領域網路的連線。使用Samba 時，SMB 通訊協定就可以在TCP/IP 通訊協定上運作，因此TCP/IP 通訊協定必須安裝在所有的用戶端上。

NetBIOS 是用來進行機器之間通訊的軟體介面(API)。在此提供了名稱服務。它允許連線至網路的機器保留自己的名稱。在保留後，就可以使用名稱來定址這些機器。在此沒有檢查名稱的中央程序。在網路上的任何機器都可以保留它所需的任何數量名稱，只要這些名稱尚未使用。現在可以針對不同的網路架構實行NetBIOS 介面。有一個與網路硬體一起緊密運作的執行程序，稱為NetBEUI，不過這通常稱為NetBIOS。與NetBIOS 一起執行的網路通訊協定是Novell 的IPX (經由TCP/IP 的NetBIOS) 與TCP/IP。

經由TCP/IP所傳送的NetBIOS名稱，與/etc/hosts中所使用的名稱，或由DNS所定義的名稱完全不相同。NetBIOS使用自己完全獨立的命名慣例。不過一般建議使用與DNS主機名稱相對應的名稱，如此可使管理較為輕鬆。Samba預設是使用此對應名稱。

所有一般的作業系統(例如Mac OS X、Windows以及OS/2)都支援SMB通訊協定。TCP/IP通訊協定必須安裝在所有的電腦上。Samba提供適用於不同UNIX類別的用戶端。就Linux而言，有一個SMB的核心模組，允許在Linux系統層次上整合SMB資源。

SMB伺服器透過共用提供用戶端硬碟空間。共用包含伺服器上的目錄及其子目錄。它是利用名稱來輸出，並且可藉由其名稱來存取。共用名稱可以設成任何名稱—它並不需要是輸出目錄的名稱。也會指定一個名稱給印表機。用戶端可以透過其名稱存取印表機。

32.1 設定伺服器

如果您想要使用Samba做為伺服器，請安裝samba。以`rcnmb start && rcsmb start`啟動Samba所需的服務，並以`racsmb stop && rcnmb stop`停止它們。

Samba的主要組態檔為/etc/samba/smb.conf。這個檔案可以分成兩個邏輯部份。[global]區段包含中央與全域設定值。[share]區段包含個別檔案與印表機共用。利用此方法，就可以在[global]區段中以不同的方式或以全域方式設定關於共用的細節，它可以加強組態檔的結構透明化。

32.1.1 全域區段

下列[global]區段的參數需要做一些調整，以符合網路設定的需求，讓其他機器可以透過Windows環境中的SMB存取Samba伺服器。

workgroup = TUX-NET 這一行是將Samba伺服器指定給工作群組。以網路環境中適當的工作群組取代TUX-NET。除非您已將這個名稱指定給網路中的其他機器，否則Samba伺服器將會以其DNS名稱來顯示。如果沒有DNS名稱，請使用`netbiosname=MYNAME`設定伺服器名稱。如需關於此參數的詳細資訊，請參閱`mansmb.conf`。

os level = 2 此參數會觸發Samba伺服器是否嘗試變成其工作群組的LMB(本地主要的瀏覽器)。請選擇一個非常低的值，以使現有的Windows網路能避免設定不當的Samba伺服器所造成的任何問題。如需關於此重要主

題的詳細資訊，請參閱套件文件中textdocs 子目錄下的BROWSING.txt 與BROWSING-Config.txt 檔案。

如果没有其他的SMB 伺服器在網路中(例如，Windows NT 或2000 伺服器)，而且您想要Samba 伺服器保留在本地環境中所存在的所有系統清單，請將os level 設成更高的值(例如，65)。接著就會將Samba 伺服器選擇成本地網路的LMB。

當變更此設定值時，請小心地考慮這個值將會如何影響現有的Windows 網路環境。首先請在獨立的網路中或在一天中非重要的時間測試變更。

wins support 與wins server 若要將Samba 伺服器整合至含有主動WINS 伺服器的現有Windows 網路中，請啓用wins server 選項，並將其值設為該WINS 伺服器的IP 位址。

如果您的Windows 機器已連線至獨立的子網路中，而且應該要能夠看到彼此，則需要設定WINS 伺服器。若要將Samba 伺服器變成像這樣的WINS 伺服器，請設定wins support = Yes 選項。請確定網路中只有一個Samba 伺服器啓用了這個設定值。wins server 與wins support 選項絕不能在smb.conf 檔案中同時啓用。

32.1.2 共用

下列範例說明如何將CD-ROM 光碟機與使用者目錄(homes) 開放給SMB 用戶端使用。

[cdrom] 若要避免不小心將CD-ROM 光碟機開放成共用，請以備註符號停用這些行(在此例中為分號)。請在第一個資料欄中移除分號，以便和Samba 共用CD-ROM 光碟機。

Example 32.1: CD-ROM 共用

```
:[cdrom]
;      comment = Linux CD-ROM
;      path = /media/cdrom
;      locking = No
```

[cdrom] and comment [cdrom] 這個項目是在網路上所有的SMB 用戶端都可以看到的共用名稱。可以另外再加入一個comment，以進一步描述共用。

path = /media/cdrom path 會輸出/media/cdrom 目錄。

利用限制非常嚴格的預設組態，就可以將這種共用只開放給出現在此系統上的使用者共用。如果這個共用應該開放每個人使用，請將`guest ok = yes`加入組態。這個設定值可以將讀取權限開放給網路上的每個人使用。建議您處理此參數時必須極為小心。這將會在`[global]`區段中套用更多此參數的使用。

[homes] `[home]` 共用在這裏特別重要。如果使用者擁有Linux 檔案伺服器以及其自己主目錄的有效帳戶與密碼，就可以連線到主目錄。

Example 32.2: 主目錄共用

```
[homes]
    comment = Home Directories
    valid users = %S
    browseable = No
    read only = No
    create mask = 0640
    directory mask = 0750
```

[homes] 只要沒有其他的共用，使用共用的使用者名稱連線至SMB 伺服器，就會使用**[homes]** 共用指示來動態產生共用。共用的產生名稱與使用者名稱相同。

valid users = %S 只要成功地建立連線，就會以共用的具體名稱取代**%S**。至於**[homes]** 共用，這會與使用者的名稱相同。因此，使用者共用的存取權限僅限於使用者。

browseable = No 這個設定值讓共用在網路環境變成無形的。

read only = No 根據預設，Samba 會利用**read only = Yes** 參數，以禁止寫入任何輸出共用的權限。若要開放共用為可寫入的，請設定**read only = No** 的值，這與**writeable = Yes** 同義。

create mask = 0640 以MS Windows NT 為基礎的系統將無法理解UNIX 權限的概念，因此在建立檔案時，它們將無法指定權限。**create mask** 參數可以定義指定給新建立檔案的存取權限。這只會套用至可寫入的共用。實際上，這個設定值表示擁有者具有讀取與寫入權限，而擁有者的主要群組成員則具有讀取權限。**valid users = %S** 可以在即使群組具有讀取權限時禁止讀取權限。若想要使群組具有讀取或寫入權限，請停用**valid users = %S** 一行。

32.1.3 安全性層級

SMB 通訊協定是源自於DOS與Windows並直接考慮安全性的問題。每個共用存取權都以密碼保護。SMB 具有三個檢查權限的可能方法：

共用層級安全性(安全性= 共用)： 每個共用都必須指定密碼。每個知道此密碼的人員都具有該共用的存取權。

使用者層級安全性(安全性= 使用者)：

這個變化引用了使用者對SMB的概念。每個使用者都必須以自己的密碼註冊伺服器。在註冊後，伺服器可以視使用者名稱將存取權授與個別輸出的共用。

伺服器層級安全性(安全性= 伺服器)：

對其用戶端而言，Samba 會模擬在使用者層級模式中工作。不過，它會將所有的密碼查詢傳遞給另一個將會處理驗證的使用者層級模式伺服器。這個設定值將需要另一個參數(password server =)。

共用、使用者以及伺服器層級安全性之間的差異會套用至整個伺服器。因為無法針對伺服器組態的個別共用提供共用層級的安全性，並針對其他的共用提供使用者層級的安全性。然而，您可以針對系統上每個設定的IP位址執行個別的Samba伺服器。

在「Samba HOWTO 文件集」中可以找到關於此主題的詳細資訊。至於在一個系統上的多個伺服器，請注意interfaces與bind interfaces only選項。

如需進行Samba伺服器的簡單管理工作，也可以使用swat程式。它提供簡單的網頁介面，以便於設定Samba伺服器。在網頁瀏覽器中，開啓http://localhost:901並以root的使用者身份登入。然而，必須同時在/etc/xinetd.d/samba與/etc/services檔案中啓動swat。若要這麼做，請在/etc/xinetd.d/samba中，編輯disable一行，將它設爲disable = no。在man頁面中提供了關於swat的詳細資訊。

32.2 做爲登入伺服器的Samba

在以Windows用戶端爲主的網路中，通常會建議使用者只註冊一個有效的帳戶與密碼。這可藉由Samba伺服器來完成。在以Windows爲基礎的網路中，

這個工作是由設定成主要領域控制器(PDC)的Windows NT 伺服器來處理。在smb.conf的[global]區段中必須編輯的項目如範例32.3,“在smb.conf中的全域區段”所示。

Example 32.3: 在smb.conf中的全域區段

```
[global]
workgroup = TUX-NET
domain logons = Yes
domain master = Yes
```

如果使用加密的密碼來驗證(這是維護良好的MS Windows 9x 安裝、MS Windows NT 4.0 加上Service Pack 3 以及所有之後產品的預設值), Samba 伺服器必須能夠處理它們。在[global]區段中的encrypt passwords = yes 項目可以啓用此功能(在Samba 版本3, 這已是預設值)。除此之外, 必須準備符合Windows 加密格式的使用者帳戶與密碼。請使用smbpasswd -a name 指令來執行此動作。使用下列指令為電腦建立領域帳戶(Windows NT 領域概念所需):

Example 32.4: 設定機器帳戶

```
useradd hostname\$$
smbpasswd -a -m hostname
```

使用useradd 命令, 就會加上貨幣符號。當使用-m 參數時, smbpasswd 指令就會自動插入這個符號。加備註的組態範例(/usr/share/doc/packages/Samba/examples/smb.conf.SuSE) 包含自動化此工作的設定值。

Example 32.5: 機器帳戶的自動化設定

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m\$$
```

若要確定Samba 可以正確地執行此程序檔, 請選擇具有所需管理員權限的Samba 使用者。若要這麼做, 請選取一個使用者, 並將它加入ntadmin 群組。在此之後, 就可以使用下列指令指定Domain Admin 狀態給所有屬於此Linux 群組的使用者:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

如需關於此主題的詳細資訊, 請參閱/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf 中第12 章的「Samba HOWTO 文件集」。

32.3 使用YaST 設定Samba 伺服器

選擇由新的Samba 伺服器控制的工作群組或領域，以啟動伺服器組態。從「工作群組或領域名稱」下拉式功能表選取一個現有的工作群組/領域，或輸入新的群組/領域。在下一個步驟中，指定伺服器應該做為PDC (主要領域控制器) 或做為BDC (備份領域控制器)。

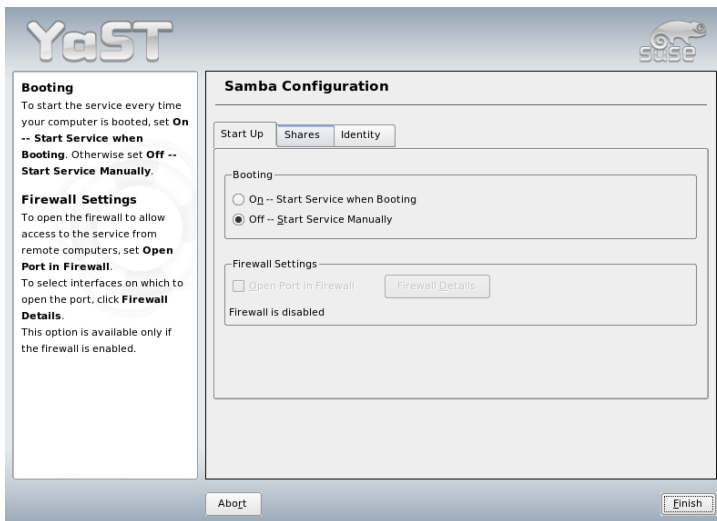


Figure 32.1: Samba 組態—啓動

在‘啓動’中啓動Samba (圖形32.1, “Samba 組態—啓動”)。使用‘在防火牆中開啓埠’以及‘防火牆細節’，以調整伺服器上的防火牆，使netbios-ns、netbios-dgm、netbios-ssn以及microsoft-ds服務的埠對所有外部與內部的介面開放，因而可確保Samba 伺服器的作業能順利進行。

在‘共用’中(圖形32.2, “Samba 組態—共用”)，決定要啓動的Samba 共用。使用‘切換狀態’以切換‘作用中’與‘非作用中’。按一下‘新增’以新增新的共用。

在‘識別’中(圖形32.3, “Samba 組態—識別”)，決定主機要關聯的領域(‘基礎設定’)以及是否要在網路中使用替代的主機名稱(‘NetBIOS 主機名稱’)。

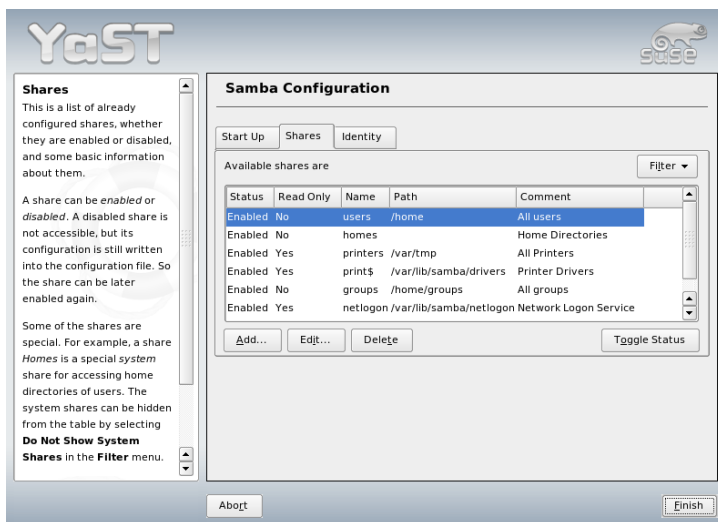


Figure 32.2: Samba 組態—共用

32.4 設定用戶端

用戶端只能透過TCP/IP存取Samba伺服器。NetBEUI與透過IPX的NetBIOS無法與Samba一起使用。

32.4.1 使用YaST設定Samba用戶端

設定Samba用戶端以存取Samba伺服器上的資源(檔案或印表機)。在'Samba工作群組'對話方塊中輸入領域或工作群組。按一下'瀏覽'以顯示可以使用滑鼠選取的所有可用的群組與領域。如果您啟動'Linux驗證也使用SMB資訊'，使用者驗證將會在Samba伺服器上執行。在完成所有的設定後，按一下'完成'以完成組態。

32.4.2 Windows 9x 與ME

Windows 9x 與ME 已經有TCP/IP的內建支援。不過，預設並不會安裝它。如果要新增TCP/IP，請進入'控制台'→'系統'中，然後選擇'新增'→'通訊協定'→

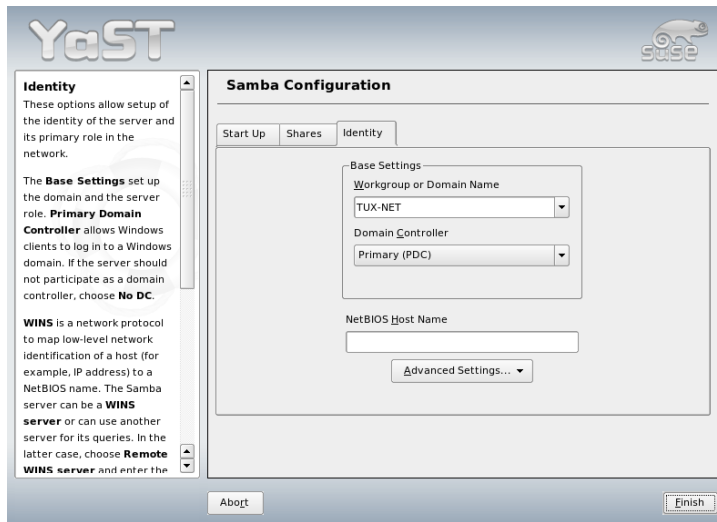


Figure 32.3: Samba 組態－識別

‘Microsoft 的 TCP/IP’。在重新啓動 Windows 機器後，按兩下網路環境中的桌面圖示以尋找 Samba 伺服器。

如果要使用 Samba 伺服器上的印表機，請從對應的 Windows 版本安裝標準或 Apple-PostScript 印表機驅動程式。最好是將此與 Linux 印表機佇列連結，它可以接受 Postscript 做為輸入格式。

32.5 最佳化

socket options 是 Samba 版本隨附的範例組態中，所提供的其中一個可能的最佳化組態。其預設的組態會參考本地的乙太網路。如需關於 socket options 的其他資訊，請參閱 smb.conf 手冊頁的相關小節，以及 socket(7) 的手冊頁。在「Samba HOWTO 文件集」的 Samba 效能調整一章中，提供了進一步的資訊。

在 `/etc/samba/smb.conf` 中的標準組態，是用來提供以Samba 團隊預設值為基礎的有用設定。不過您無法直接使用該組態，特別是在網路組態與工作群組名稱的檢視中。加備註的範例組態 `examples/smb.conf.SuSE` 包含對本地需求調整相當有用的資訊。

Samba 團隊所提供的「Samba HOWTO 文件集」，包含疑難排解一節。除此之外，文件的第五部份提供檢查組態的逐步指南。

Proxy 伺服器：Squid

Squid 是Linux 與UNIX 平台普遍使用的Proxy 快取。本節討論其組態、執行它所需的設定、如何設定系統以執行操作順暢的Proxy 設定、如何使用程式說明來收集關於使用快取的統計資料(像是Calamaris 與cachemgr) 以及如何使用squidGuard 來過濾網站內容。

33.1 做為Proxy 快取的Squid

Squid 可做為Proxy 快取。它會將物件要求從用戶端(在此例中是從網頁瀏覽器)重新導向至伺服器。當從伺服器而來的要求物件到達時，它會將物件傳送到用戶端，並在硬碟快取中保留它們的複本。快取的其中一個優點為，當有數個用戶端要求相同的物件時，都可以從硬碟快取來提供。這可讓用戶端比從網際網路更快地擷取資料。這個程序也可以減少網路流量。

除了實際快取之外，Squid 提供更廣泛的功能，例如分散在Proxy 伺服器中互相通訊階層的負荷、為所有存取Proxy 的用戶端定義更嚴格的存取控制清單，允許或拒絕使用其他應用程式來存取特定網頁，以及產生經常瀏覽的網頁統計資料，以評估使用者的瀏覽習慣。Squid 不是一般的Proxy。一般而言，它只會代理HTTP 連線。它也支援FTP、Gopher、SSL 以及WAIS 等通訊協定，但是它並不支援其他的網際網路通訊協定，例如Real Audio、新聞或視訊會議。因為Squid 只支援UDP 通訊協定提供不同快取之間的通訊，許多其他的多媒體程式並不支援。

33.2 關於Proxy 快取的說明

33.2.1 Squid 以及安全性

它也有可能使用Squid 加上防火牆，以便使用Proxy 快取來保護內部網路不受到外部的存取。防火牆將會拒絕所有的用戶端存取Squid 以外的外部服務。所有的網路連線都必須經由Proxy 來建立。

如果防火牆包含DMZ，Proxy 應該在此區域中操作。在此例中，在DMZ 中的所有電腦都必須將其記錄檔傳送到安全網路內的主機上。執行操作順暢的Proxy 的可能性會在 節33.6, “設定操作順暢的Proxy” 小節中加以描述。

33.2.2 多個快取

可以使用在Proxy 之間交換物件的方式來設定數個Proxy。這可減少總系統的負荷，並可增加在本地網路中找到已存在物件的機會。它也可以設定快取階層，使快取可以將物件要求轉送至同層級的快取或上層快取— 這將使其從本地網路的另一個快取或直接從來源取得物件。

為快取階層選擇適當的拓樸是非常重要的事，因為這樣它就不會增加網路的整體流量。就大型的網路而言，就非常合適為每個子網路設定Proxy 伺服器，並將它們連線至上層的Proxy，這樣就可以連線至ISP 的Proxy 快取。

這些通訊都是由在UDP 通訊協定最上層執行的ICP (網際網路快取通訊協定，Internet Cache Protocol) 所處理。在快取之間的資料傳輸是使用以TCP 為基礎的HTTP (超文字傳輸通訊協定，Hypertext Transmission Protocol) 來處理。

為了能找到取得物件最適合的伺服器，某個快取會將ICP 要求傳送到所有同層級的Proxy。如果有偵測到物件，就會透過具有HIT 代碼的ICP 回應來回覆這些要求，或者如果没有偵測到物件，就會透過具有MISS 代碼的ICP 回應來回覆這些要求。如果找到多個HIT 回應，Proxy 伺服器就會視某些因素來決定要下載的伺服器，例如哪個快取可以傳送最快的回覆或是哪個伺服器距離最近。如果没有收到符合的回應，則會將要求傳送到上層快取。

為了避免在網路中有不同快取的物件重複，會使用其他的ICP 通訊協定。例如CARP (快取陣列路由通訊協定，Cache Array Routing Protocol) 或是HTCP (超文字快取通訊協定，Hypertext Cache Protocol)。在網路中維護愈多的物件，則找到所需物件的機率也就愈大。

33.2.3 快取網際網路物件

並非在網路中所有可用的物件都是靜態。其中有許多動態產生的CGI頁面、訪客計數器以及加密的SSL內容文件。諸如此類的物件是不會被快取的，因為每次存取它們時，它們都會改變。

還有一個問題就是，所有儲存在快取中的其他物件，應該停留在快取中多長的時間。為了決定該停留多久，會指定各種可能的狀態給快取中的所有物件。網路以及Proxy伺服器是藉由新增標題至這些物件來找出物件的狀態，例如「上次修改」或「過期」以及對應日期。也會使用不應該快取指定該物件的其他標題。

一般而言會取代在快取中的物件，這是因為缺乏磁碟空間，而使用像是LRU(最近使用的)等演算法之故。基本上這表示Proxy會清空那些最久沒有被要求的物件。

33.3 系統需求

最重要的事情是要決定系統必須承受的最大負載。因此，特別注意負載尖峰是很重要的，因為這些尖峰有可能是每天平均值的四倍。當有疑問時，最好高估系統的需求，因為如果讓Squid在接近其容量限制下工作，有可能會造成服務品質的嚴重損失。接下來的小節將依重要順序指出系統因素。

33.3.1 硬碟

速度在快取處理過程中扮演很重要的角色，因此應該特別注意這個因素。對硬碟而言，此參數可稱為「隨機搜尋時間」，以毫秒為單位。因為Squid所讀取或寫入硬碟的資料區塊通常都相當的小，所以硬碟的搜尋時間比其資料輸送量還要重要。對Proxy的用途而言，具有高旋轉速的硬碟可能是較好的選擇，因為它們允許以更快的速度將讀寫頭放置在所需的位置。今日快速的SCSI硬碟的搜尋時間為四毫秒以下。有一個方法可能可以增加系統的速度，就是同時使用多個磁碟或是運用分段RAID陣列。

33.3.2 磁碟快取的大小

在小的快取中，HIT(尋找要求的物件已在該處)的機率是很小的，因為該快取很容易就會被填滿，所以較新的物件取代要求物件的機率較小。例如，如果快取有1GB可用，而且使用者一天只瀏覽10MB，則需要一百天以上才能將快取填滿。

決定所需快取大小的最簡易方法為考慮連線的最大傳輸速率。每秒1百萬位元的連線，其最大的傳輸速率為每秒125 KB。如果所有的流量最後都在快取中，則1個小時就會新增至450 MB，假設所有的流量都只在8個工作小時中產生，則一天就可達到3.6 GB。由於連線一般都不會用到其容量上限，因此可以假設快取所處理的總資料容量大約為2 GB。這是為什麼在Squid的範例中需要2 GB的磁碟空間，以快取一天中已瀏覽的資料量。

33.3.3 RAM

Squid 所需的記憶體量(RAM) 與快取中的物件數目直接相關。Squid 也可以將快取物件的參照以及常要求的物件儲存在主記憶體中以加速此資料的擷取速度。隨機存取記憶體比硬碟的速度快很多。

除此之外，Squid 需要在記憶體中保留其他的資料，例如所有已處理IP位址的表格、精確的領域名稱快取、最常要求的物件、存取控制清單、緩衝區等等。

讓Squid處理程序擁有足夠的記憶體是非常重要的，因為若使用磁碟進行交換，系統效能就會大幅地降低。cachemgr.cgi工具可用於快取記憶體管理。將會在節33.7, "cachemgr.cgi" 小節中介紹此工具。

33.3.4 CPU

Squid 不是一個需要密集使用CPU的程式。只有在載入或檢查快取的內容時，才會增加處理器的負載。使用多處理器的機器並不能提升系統的效能。若要提升效率，最好買更快的磁碟或新增更多的記憶體。

33.4 啟動Squid

Squid 已預先設定在SUSE LINUX 中，所以您可以在安裝後立即啟動它。若要確保更平順的啟動，應該使用至少一個名稱伺服器，以及可連線的網際網路來設定網路。如果撥接連線是使用動態DNS組態，就有可能產生問題。在這個例子中，至少應該明確地輸入名稱伺服器，因為在/etc/resolv.conf中，如果沒有偵測到DNS伺服器，就不會啟動Squid。

33.4.1 開始和停止Squid的指令

若要啟動Squid，請以root的身份在指令行中輸入`rcsquid start`。在第一次啟動時，必須在`/var/squid/cache`中定義目錄結構。定義目錄可由自動啓

動/etc/init.d/squid 程序檔來完成，這可能需要花費數秒鐘或甚至幾分鐘才能完成。如果done以綠色出現在右邊，則表示已成功載入Squid。若要在本地系統上測試Squid的功能，請在瀏覽器中輸入localhost 做為Proxy，並輸入3128 做為埠。

若要允許所有的使用者透過網際網路存取Squid，請將/etc/squid/squid.conf 組檔態中的http_access deny all 項目變更為http_access allow all。然而，當您這麼做時，請考慮到此動作將使Squid可供任何人完全存取。因此，請定義控制Proxy存取的ACL。在節33.5.2, “存取控制的選項” 小節中提供了關於此的詳細資訊。

在修改/etc/squid/squid.conf 組態檔後，Squid 必須重新載入組態檔。請以rcsquid reload 執行此動作。或者也可以使用rcsquid restart 完全重新啓動Squid。

rcsquid status 指令可用來檢查Proxy 是否正在執行。rcsquid stop 指令會造成Squid 關閉。這可能需要花一段時間，因為Squid 會在中斷與用戶端的連線並將其資料寫入磁碟前，先等待半分鐘(在/etc/squid/squid.conf 中的shutdown_lifetime 選項)。

┌

終止Squid

使用kill 或killall 來終止Squid 有可能會損毀快取。若要能夠重新啓動Squid，就必須先刪除損毀的快取。

└

如果Squid 在成功啓動後，仍然於一小段時間後即停止運作，請檢查是否有錯誤名稱的伺服器項目，或者是否缺少/etc/resolv.conf 檔案。Squid 會在/var/squid/logs/cache.log 檔案中記錄啓動失敗的原因。如果在系統開機時，應該自動載入Squid，請為所需的Runlevel 使用YaST Runlevel 編輯器以啓動Squid。

Squid 的解除安裝不會移除記錄檔的快取階層。若要移除這些階層，請手動刪除/var/cache/squid 目錄。

33.4.2 本地DNS 伺服器

即使本地DNS 伺服器不管理自己的領域，也可以設定本地DNS 伺服器。它可以做為僅供快取的名稱伺服器，也可以透過root 名稱伺服器來解決DNS 要求，而不需任何特殊的組態。如果您在/etc/resolv.conf 檔案中為本地DNS 伺服器輸入localhost 的127.0.0.1 IP 位址，當Squid 啓動時應該會找到有效的名稱

伺服器。在forwarders 下的/etc/named.conf 組態檔中輸入提供者的名稱伺服器以及其IP 位址。不過，如果您有執行防火牆，請確定DNS 要求可以通過防火牆。

33.5 /etc/squid/squid.conf 組態檔案

所有的Squid Proxy 伺服器的設定值都是在/etc/squid/squid.conf 檔案中設定。在第一次啓動Squid 時，在此檔案中不需做任何變更，但是外部用戶端一開始爲拒絕存取。Proxy 可供localhost 使用。預設的埠是3128。預先安裝的/etc/squid/squid.conf 提供關於選項以及許多範例的詳細資訊。幾乎所有的項目都是以# (加備註的行) 開始，而且可以在行的結尾找到相關的指定。指定的值永遠都與預設值相關，因此在大部份情況下，移除備註符號而未變更任何參數，實際上沒有什麼效果。如果有可能，請保留原始的範例，並在行的下方插入選項以及修改過的參數。以這種方法即可輕易地解譯預設值及變更。

在更新後調整組態檔案

如果您是從較早的Squid 版本更新，建議您編輯新的/etc/squid/squid.conf，並且只套用在舊檔案中所做的變更。如果您嘗試執行舊的squid.conf，請注意該組態有可能無法再運作，因爲選項有時會被修改並增加新的變更。

33.5.1 一般組態選項(選擇)

http_port 3128 這是Squid 爲用戶端要求所傾聽的埠。預設的埠是3128，但是8080 也是常用的埠。如果有需要，請指定數個以空白分隔的埠號碼。

cache_peer (hostname) (type) (proxy-port) (icp-port)

在此您可以輸入上層Proxy，例如如果您想要使用ISP 的Proxy。針對(hostname) 的部份，輸入要使用的Proxy IP 位址，並針對(type) 部份，輸入parent。至於(proxy-port) 的部份，則輸入上層運算子也會設定的埠號碼，以利在瀏覽器中使用，通常是8080。如果上層的ICP 埠是未知的，而且其用途與提供者無關，請將(icp-port) 設爲7 或0。除此之外，在埠號碼禁止使用ICP 通訊協定後，應該指定default 與no-query。就提供者Proxy 而言，Squid 接著就會像一般的瀏覽器一樣地運作。

cache_mem 8 MB 這個項目定義Squid可以使用的快取記憶體量。預設值為8 MB。

cache_dir ufs /var/cache/squid/ 100 16 256

cache_dir 這個項目是定義所有儲存在磁碟上的物件之目錄。在結尾處的數目代表可以使用的最大磁碟空間(MB)，以及在第一層與第二層的目錄數目。*ufs* 參數應該獨立存在。*/var/cache/squid* 目錄預設是佔用磁碟空間100 MB，而且可以在其中建立16個子目錄，另外每個目錄還包含256個子目錄。當指定要使用的磁碟空間時，請保留足夠的可用磁碟空間。在此指定可用磁碟空間的50% (最小) 至80% (最大) 為合理的值。最後兩個指定目錄的數目應該謹慎地增加，因為太多的目錄也有可能導致效能降低。如果您有數個共用快取的磁碟，請輸入數行*cache_dir*。

cache_access_log /var/log/squid/access.log

記錄訊息的路徑。

cache_log /var/log/squid/cache.log 記錄訊息的路徑。

cache_store_log /var/log/squid/store.log

記錄訊息的路徑。

這三個項目可以指定Squid記錄其所有動作的路徑。一般而言，這裏不會有所變更。如果Squid負荷過重，則將快取與記錄檔分散到數個磁碟會是一個好辦法。

emulate_httpd_log off 如果該項目是設為*on*，就會取得可以讀取的記錄檔。然而有些試用程式無法解譯此檔案。

client_netmask 255.255.255.255 使用這個項目時，在記錄檔中的遮罩IP位址就可以隱藏用戶端的身分。如果您在這裏輸入255.255.255.0，IP位址的最後一位數就會設為0。

ftp_user Squid@ 使用這個項目設定匿名FTP登入應該使用的Squid密碼。在此也可以指定有效的電子郵件地址，因為某些FTP伺服器會檢查這些地址的有效性。

cache_mgr webmaster 如果Squid意外損毀，Squid應傳送訊息給此電子郵件地址。預設值為網站管理員。

logfile_rotate 0 如果您執行*squid -k rotate*，則Squid會輪換安全的記錄檔。在此程序中會計算檔案的數目，而且在到達指定的值後，就會覆寫最舊的檔案。預設值為0，因為歸檔和刪除在SUSE LINUX中的記錄檔是由設定在*/etc/logrotate/squid* 組態檔中的*cron*工作所執行。

append_domain *domain*;
使用 *append_domain*，可在沒有指定領域時，指定要自動附加的領域。通常，可在此處輸入自己的領域，因此在瀏覽器中輸入 *www* 就可以存取您自己的網頁伺服器。

forwarded_for *on*
如果您將項目設為 *off*，則 Squid 會移除 IP 位址以及從 HTTP 要求而來的用戶端系統名稱。

negative_ttl 5 minutes; **negative_dns_ttl** 5 minutes

一般而言，您不需要變更這些值。不過，就算您有撥號連線，但是網際網路有時可能無法使用。雖然已重新建立網際網路連線，Squid 仍然會記錄失敗的要求，並拒絕發出新的要求。像這個例子中，將「分鐘」變更為「秒」，然後在瀏覽器中按一下「重新載入」後，幾秒鐘之後就會進行撥號程序。

never_direct *allow* *<acl_name>*
為了防止 Squid 直接從網際網路接受要求，請使用上述的指令以強制連線至另一個 Proxy。此指令應該在之前已於 *cache_peer* 中輸入。如果 *all* 是指定為 *<acl_name>*，請強制將所有的要求直接轉送至 *parent*。例如，如果您所使用的提供者，嚴格地規定其 Proxy 的使用方式或拒絕其防火牆直接存取網際網路，就可能需要執行此動作。

33.5.2 存取控制的選項

Squid 提供控制 Proxy 存取的詳細系統。透過執行 ACL，可以輕鬆並完整地設定。這牽涉到使用連續處理規則的清單。在使用 ACL 前必須先定義它。某些預設的 ACL，例如 *all* 與 *localhost* 已經存在。然而，僅定義 ACL 並不代表它實際上已被套用。這只會發生在連接 *http_access* 規則使用時。

acl *acl_name*;
type;
data;
至少需要三種規格才能定義 ACL。可以任意選擇 *acl_name* 的名稱。至於 *type* 的部份，可從各種不同的選項中選取(可在 */etc/squid/squid.conf* 檔案中的 *ACCESS CONTROLS* 小節中找到)。*data* 的規格是視個別的 ACL 類型而定，而且也可以從檔案讀取，例如，透過主機名稱、IP 位址或 URL。下列是一些簡單的範例：

```
acl mysurfers srcdomain .my-domain.com
acl teachers src 192.168.1.0/255.255.255.0
acl students src 192.168.7.0-192.168.9.0/255.255.255.0
acl lunch time MTWHF 12:00-15:00
```

http_access *allow* *acl_name*;
http_access 定義哪些人可使用 Proxy，以及哪些人可存取網際網路的哪些內容。因此，必須指定 ACL。已在上述定義的 *localhost* 與 *all*，可透過 *deny* 或 *allow* 拒絕或允許存取。您可以建立包含任

何數目的`http_access` 項目之清單，從上至下處理並視哪個項目先發生，以允許或拒絕存取個別的URL。最後一個項目應該永遠為`http_access deny all`。在下列範例中，`localhost` 對於每個項目都擁有完整的存取權，而所有其他的主機則完全無法存取。

```
http_access allow localhost
http_access deny all
```

在另一個使用這些規則的範例中，`teachers` 群組永遠具有網際網路的存取權。`students` 群組只能取得在星期一到星期五午休時間的存取權。

```
http_access deny localhost
http_access allow teachers
http_access allow students lunch time
http_access deny all
```

基於可讀性的理由，具有`http_access` 項目的清單，只能在`/etc/squid/squid.conf` 檔案中的指定位置中輸入。也就是，在文字之間

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS
```

以及最後

```
http_access deny all
```

redirect_program /usr/bin/squidGuard

使用此選項，就可以指定如`squidGuard` 之類的新導向器，來封鎖不需要的URL。使用Proxy 驗證以及適當的ACL，就可以為不同的使用者群組個別地控制網際網路的存取。`squidGuard` 是可以安裝和設定的獨立套件。

auth_param basic program /usr/sbin/pam_auth

如果使用者必須在Proxy 上驗證，請設定對應的程式，例如`pam_auth`。當第一次存取`pam_auth` 時，使用者會看到輸入使用者名稱與密碼的登入視窗。除此之外，仍然需要ACL，因此只有擁有有效登入的用戶端可以使用網際網路：

```
acl password proxy_auth REQUIRED

http_access allow password
http_access deny all
```

在`proxy_auth` 後面的`REQUIRED`，可以使用允許的使用者名稱清單來取代，或是以這類清單的路徑來取代。

`ident_lookup_access allow jacl_name;`

使用這個項目，就會為所有ACL定義的用戶端執行`ident`要求以尋找每個使用者的身份。如果您將`all`套用至`jacl_name;`，這對所有的用戶端都是有效的。另外，`ident`精靈必須在所有的用戶端上執行。在Linux上，請為此用途安裝`pidentd`套件。在Windows上，可以從網際網路下載可用的免費軟體。若要確保只允許`ident`查詢為成功的用戶端，請在此定義對應的ACL：

```
acl identhosts ident REQUIRED
```

```
http_access allow identhosts  
http_access deny all
```

此處，也可使用允許的使用者名稱清單來取代`REQUIRED`。使用`ident`會大幅增加存取時間，因為每個要求都會重複`ident`查詢。

33.6 設定操作順暢的Proxy

使用Proxy伺服器的一般方式如下：網頁瀏覽器會將要求傳送至Proxy伺服器中的某個埠，而且Proxy會提供這些需要的物件，不論它們是否在其快取中。使用網路工作時，可能會發生數種狀況：

- 基於安全理由，建議所有的用戶端都使用Proxy瀏覽網際網路。
- 所有的用戶端都必須使用Proxy，不論它們是否注意到它。
- 雖然Proxy在網路上是變動的，但現有的用戶端應該保留其舊有的組態。

在所有的例子中，都可使用操作順暢的Proxy。其原則非常容易：Proxy會攔截和回應網頁瀏覽器的要求，因此網頁瀏覽器不需要知道頁面的來源處，即可收到要求的頁面。如名稱所示，整個程序會流暢地執行。

33.6.1 核心組態

首先，請確定Proxy伺服器的核心，支援操作順暢的Proxy。如果不支援的話，請將這些選項新增至核心並重新編譯它。如需詳細資訊，請參閱 章9, Linux 核心小節。

33.6.2 在/etc/squid/squid.conf 中的組態選項

在/etc/squid/squid.conf 檔案中啓動選項，以啓動和執行操作順暢的Proxy：

- `httpd_accel_host virtual`
- `httpd_accel_port 80` # 實際HTTP 伺服器所在的埠號碼。
- `httpd_accel_with_proxy on`
- `httpd_accel_uses_host_header on`

33.6.3 使用SuSEfirewall2 的防火牆組態

現在，請使用埠轉送規則的協助，透過防火牆將所有接收的要求，重新導向至Squid 埠。若要這麼做，請使用隨附的SuSEfirewall2 工具。在/etc/sysconfig/SuSEfirewall2 中可以找到其組態檔。組態檔是由有依據的項目所組成。即使只是設定操作順暢的Proxy，仍然必須設定一些防火牆選項：

- 指向網際網路的裝置：`FW_DEV_EXT="eth1"`
- 指向網路的裝置：`FW_DEV_INT="eth0"`

在允許存取不受信任網路(如網際網路)的防火牆上，設定埠與服務(請參閱/etc/services)。在此範例中，僅提供對外的Web 服務：

```
FW_SERVICES_EXT_TCP="www"
```

在允許存取安全網路的防火牆上，定義埠或服務(請參閱/etc/services)，以及TCP 與UDP 服務：

```
FW_SERVICES_INT_TCP="domain www 3128"  
FW_SERVICES_INT_UDP="domain"
```

這裡允許存取Web 服務與Squid (其預設埠為3128)。“domain” 服務代表DNS (領域名稱服務)。這個服務受到普遍的使用。否則，請直接將它從上述項目移除，並將下列選項設為no：

```
FW_SERVICE_DNS="yes"
```

最重要的選項是數字15：

Example 33.1: 防火牆組態：選項15

```
#
# 15.)
# Which accesses to services should be redirected to a local port
# on the firewall machine?
#
# This can be used to force all internal users to surf via your
# Squid proxy, or transparently redirect incoming Web traffic to
# a secure Web server.
#
# Choice: leave empty or use the following explained syntax of
# redirecting rules, separated with spaces.
# A redirecting rule consists of 1) source IP/net,
# 2) destination IP/net, 3) original destination port and
# 4) local port to redirect the traffic to, separated by a colon,
# e.g. "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"
#
```

上方的註解顯示要遵照的語法。首先，輸入存取Proxy 防火牆的內部網路之IP位址與網路遮罩。其次，輸入這些用戶端傳送其要求的IP位址與網路遮罩。如果是網頁瀏覽器，請將網路指定為0/0，萬用字元表示「可到任何位置」。之後，輸入這些要求所傳送的原始目的埠，最後，輸入重新導向所有這些要求的目的埠。由於Squid支援HTTP以外的通訊協定，請將其他埠的要求重新導向Proxy，例如FTP(埠21)、HTTPS、或SSL(埠443)。在此範例中，Web服務(埠80)會重新導向至Proxy埠(埠3128)。如果需要新增更多的網路或服務，必須在各個項目中以空格隔開。

```
FW_REDIRECT_TCP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
FW_REDIRECT_UDP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
```

若要啓動防火牆並使用它所包含的新組態，請在/etc/sysconfig/SuSEfirewall12 檔案中變更項目。START_FW 項目必須設為"yes"。

啓動Squid，如節33.4, "啓動Squid" 小節所示。若要檢查每個項目是否能正常運作，請檢查在/var/log/squid/access.log 中的Squid記錄。

若要確認是否已正確設定所有的埠，請從網路之外的任何電腦執行機器上的埠掃描。只開啓Web服務(埠80)。若要以nmap掃描埠，指令語法為nmap -O IP_address。

33.7 cachemgr.cgi

快取管理員(cachemgr.cgi)是一種CGI公用程式，用來顯示關於執行Squid程序的記憶體使用率之統計資料。它也是管理快取和檢視統計資料較方便的方式，因為不需要登入伺服器。

33.7.1 設定

首先，需要在系統上執行網頁伺服器。若要檢查Apache是否已在執行，請以root的身份輸入`rcapache status`指令。如果出現像這類的訊息：

```
Checking for service httpd: OK
Server uptime: 1 day 18 hours 29 minutes 39 seconds
```

請確定Apache正在機器上執行。否則，請輸入`rcapache start`以使用SUSE LINUX預設值啟動Apache。設定Apache的最後一個步驟是將`cachemgr.cgi`檔案複製到Apache的`cgi-bin`目錄：

```
cp /usr/share/doc/packages/squid/scripts/cachemgr.cgi /srv/www/cgi-bin/
```

33.7.2 在`/etc/squid/squid.conf`中的快取管理員ACL

快取管理員需要原始檔案中某些預設的設定值：

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

使用下列規則：

```
http_access allow manager localhost
http_access deny manager
```

第一個ACL是最重要的，因為快取管理員會嘗試透過`cache_object`通訊協定與Squid通訊。

下列規則是假設網頁伺服器與Squid是在相同的機器上執行。假設在快取管理員與Squid之間的通訊是源自於另一個電腦上的網頁伺服器(包含範例33.2,“存取規則”範例中的額外ACL)。

Example 33.2: 存取規則

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webserver src 192.168.1.7/255.255.255.255 # webserver IP
```

接著在範例範例33.3, “存取規則” 中新增規則。

Example 33.3: 存取規則

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

設定管理員的密碼以存取更多的選項，如遠端關閉快取或是檢視快取的詳細資訊。因此，請以管理員的密碼以及要檢視的選項清單設定 `cachemgr_passwd` 項目。這個清單會顯示成 `/etc/squid/squid.conf` 中項目備註的一部份。

每次變更組態檔時，請重新啓動Squid。以 `rscsquid reload` 輕鬆地執行此動作。

33.7.3 檢視統計資料

瀏覽對應的網站— <http://webserver.example.org/cgi-bin/cachemgr.cgi>。按‘繼續’，然後瀏覽不同的統計資料。快取管理員所顯示每個項目的詳細資訊，可在Squid FAQ中找到，網址為：<http://www.squid-cache.org/Doc/FAQ/FAQ-9.html>。

33.8 squidGuard

本小節並不是用來說明squidGuard的廣泛組態，而是簡單的介紹並給予使用上的一些建議。如需更深入的組態問題，請參閱squidGuard網站，網址為<http://www.squidguard.org>。

squidGuard 是一種免費(GPL)、有彈性及快速的過濾器，也是重新導向器以及Squid的存取控制器插件。它可讓您在Squid快取上，針對不同的使用者群組，使用不同的限制來定義多重存取規則。squidGuard使用Squid的標準重新導向器介面。

squidGuard 可以執行下列動作：

- 將某些使用者的Web存取權，限制為已接受的清單或已知的網頁伺服器或URL。
- 針對某些使用者，封鎖某些列示或列為黑名單的網頁伺服器或URL的存取權。
- 針對某些使用者，封鎖符合一般運算式或文字清單的URL。
- 將封鎖的URL，重新導向至「智慧型」的CGI資訊頁面。
- 將未註冊的使用者重新導向至註冊表單。
- 將橫幅重新導向至空白的GIF。
- 根據時間、星期、日期等，使用不同的存取規則。
- 針對不同的使用者群組，使用不同的規則。

squidGuard 與Squid 無法用於：

- 編輯、過濾或審查文件內的所有文字。
- 編輯、過濾或審查HTML內嵌的程序檔語言，例如JavaScript或VBscript。

使用前，請先安裝squidGuard。以/etc/squidguard.conf提供最小的組態檔。尋找在<http://www.squidguard.org/config/>中的組態範例。稍後請使用較複雜的組態設定值來測試。

接著，如果用戶端要求已列為黑名單的網站，請建立虛擬的「拒絕存取」頁面，或者建立一些複雜的CGI頁面以重新導向Squid。強烈建議使用Apache。

現在，請將Squid設定為使用squidGuard。在/etc/squid.conf檔案中使用下列項目：

```
redirect_program /usr/bin/squidGuard
```

另一個名稱為redirect_children的選項，可以設定在機器上執行的「重新導向」程序數目(在此例中為squidGuard)。squidGuard的速度相當快，可處理許多要求：在具有5,900個領域以及7,880個URL(總計13,780)的500 MHz Pentium上，10秒之內可以處理100,000個要求。因此，不建議設定四個以上的程序，因為這些程序的配置將會消耗相當大的的記憶體量。

```
redirect_children 4
```

最後，請執行`rcsquid reload`，讓Squid載入新的組態。現在，請使用瀏覽器測試您的設定值。

33.9 使用Calamaris 產生快取報告

Calamaris 是一種Perl 程序檔，用來產生ASCII 或HTML 格式的快取活動報告。它會使用原始的Squid 存取記錄檔。Calamaris 首頁是位於<http://Calamaris.Cord.de/>。該程式非常容易使用。

以root 的身份登入，然後輸入`cat access.log.files | calamaris <options> > reportfile`。當以管道來排列一個以上的記錄檔時，會依時間順序將較舊的檔案排在前面來排列記錄檔。以下是程式的某些選項：

- a 輸出所有可用的報告
- w 以HTML 報告輸出
- l 在報告標題中包含訊息或標誌

使用`man calamaris` 在程式的手冊頁中，可以找到各種選項的詳細資訊。

以下是典型的範例：

```
cat access.log.2 access.log.1 access.log | calamaris -a -w \  
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

這會將報告放在網頁伺服器的目錄中。需要有Apache 才能檢視報告。

另一個強大的快取報告產生器工具為SARG (Squid 分析報告產生器)。如需更多的詳細資訊，請參閱：<http://web.onda.com.br/orso/>。

33.10 詳細資訊

請瀏覽Squid 的首頁，網址為<http://www.squid-cache.org/>。此處可以找到「Squid 使用者指南」(Squid User Guide) 以及有關Squid 常見問題集(FAQ) 的豐富資訊。

在安裝後，可以在howtoen 中找到有關操作流暢的Proxy 之howto 資訊，檔案名稱為[/usr/share/doc/howto/en/txt/TransparentProxy.gz](#)。除此之外，可以在squid-users@squid-cache.org 中找到Squid 的可用郵件清單。這個的歸檔是位於<http://www.squid-cache.org/mail-archive/squid-users/>。

Part IV

管 理

Linux 的安全性

偽裝及防火牆可確保資料流及資料交換受到控制。SSH (保全外圍程序，Secure Shell) 可確保您透過加密的連線登入遠端主機。加密檔案或整個分割區能夠保護您的資料，不讓第三方存取您的系統。除了這些純技術性的說明外，最後一節還專門討論了Linux 網路的安全性層面。

34.1	偽裝與防火牆	502
34.2	SSH：安全網路作業	510
34.3	加密分割區和檔案	514
34.4	安全性與機密性	516

34.1 偽裝與防火牆

只要Linux用於網路環境，您可以使用允許操作網路封包的核心功能，讓內部和外部網路區域之間保持區隔。Linux netfilter 架構提供建立有效防火牆的方法，將不同的網路分開。利用iptables (規則集定義的一般表格結構) 的幫助，準確地控制允許傳遞網路介面的封包。這類封包過濾器只要透過SuSEfirewall2 及對應YaST 模組的協助即可輕鬆設定。

34.1.1 使用iptables 過濾封包

netfilter 與iptables 元件負責過濾及操作網路封包，以及網路位址轉譯(NAT, Network Address Translation)。過濾標準和任何與其關聯的動作以鏈結方式儲存，在個別網路封包抵達時必須相互符合。符合的鏈結儲存於表格內。iptables 指令允許您變更這些表格和規則集。

Linux 核心維護三個表格，每個針對封包過濾器的特定功能種類：

過濾器 此表格保存一組過濾器規則，因為它以更嚴格的方式執行「封包過濾」(*packet filtering*) 機制，例如，決定封包是否允許通過(ACCEPT) 或放棄(DROP)。

nat 此表格定義對封包之來源和目標位址的任何變更。使用這些功能也可以讓您執行「偽裝」(*masquerading*)，這是NAT 用來連結私有網路與網際網路的一種特殊方式。

mangle 此表格中的規則可以操作儲存於IP 標頭中的值(如服務類型)。

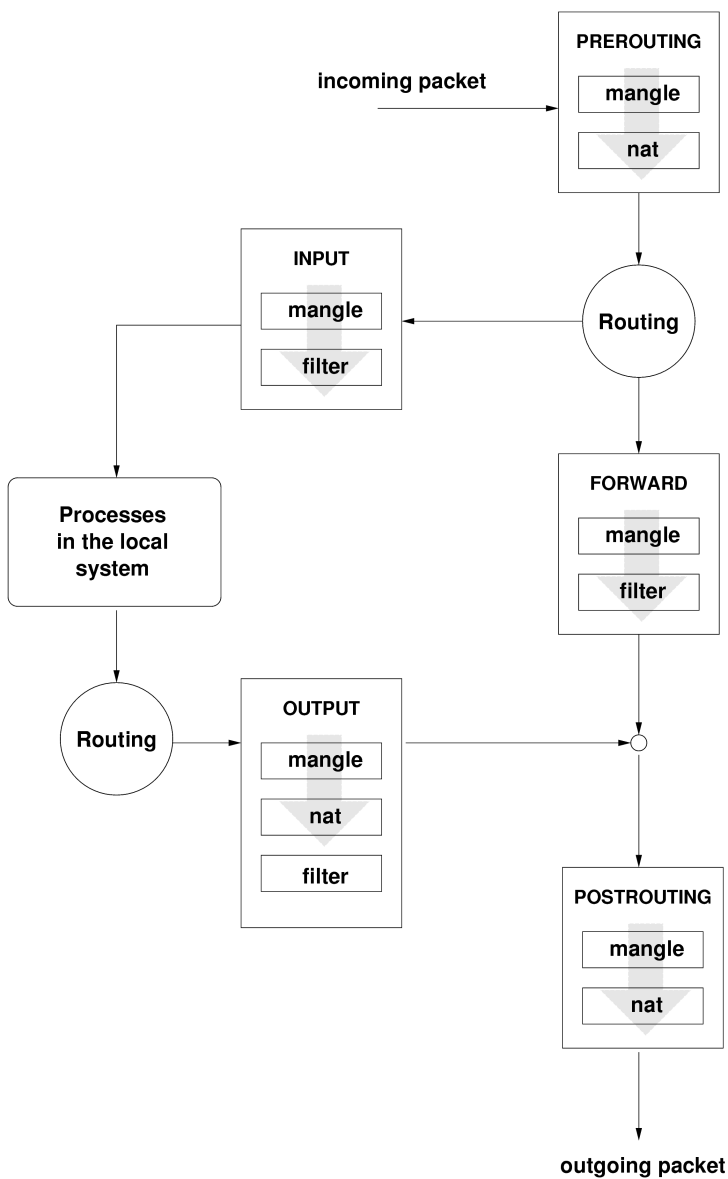


Figure 34.1: iptables : 封包的可能路徑

上述表格包含數個符合封包的預先定義鏈結：

PREROUTING 此鏈結套用到內送封包。

INPUT 此鏈結套用到預定送到系統內部程序的封包。

FORWARD 此鏈結套用到僅透過系統路由的封包。

OUTPUT 此鏈結套用到來自於從系統本身的封包。

POSTROUTING 此鏈結套用到所有外送封包。

圖形34.1, “iptables：封包的可能路徑” 說明網路封包在指定系統上傳送的路徑。爲了簡化的緣故，圖中將表格列爲鏈結的各部份，但是實際上，這些鏈結是在表格本身內。

所有可能情況中最簡單的一種，系統本身預定的內送封包抵達eth0 介面。封包先參照mangle 表的PREROUTING 鏈結，然後再參照nat 表的PREROUTING 鏈結。下一個步驟有關封包的路由，決定封包的實際目標是系統本身的程序。傳遞mangle 與filter 表的INPUT 鏈結後，封包最後會到達其目標，也就是確定符合filter 表的規則。

34.1.2 偽裝基本原則

偽裝是NAT (網路位址轉譯) 的Linux 專用形式。它可以用來連結小型LAN (主機使用私人範圍的IP 位址；請參閱節22.1.2, “網路遮罩與路由”) 與網際網路(使用正式IP 位址)。若要讓LAN 主機能夠連接網際網路，這些主機的私人位址會轉譯爲正式位址。這是在路由器上完成的，路由器當作LAN 與網際網路之間的閘道。基本原則很簡單：路由器擁有一個以上的網路介面，通常是網路卡以及另一個與網際網路連結的介面。後者連結路由器與外界，一或多個其他的則連結路由器與LAN 主機。當區域網路中的這些主機連接到路由器的網路卡時(如eth0)，它們可以傳送任何不是預定給區域網路的封包到其預設的閘道或路由器。

使用正確的網路遮罩

設定您的網路時，確定廣播位址及網路遮罩在所有本地主機上都是相同的。如果不是，會造成網路中斷，因爲封包無法正確傳送。

如所述，只要其中一個LAN 主機傳送預定給網際網路位址的封包時，就會送到預設路由器。不過，路由器必須先設定才能轉遞這類封包。爲了安全的緣

故，SUSE LINUX 在預設安裝時不會啓用此功能。若要啓用，將檔案/etc/sysconfig/sysctl 中的變數IP_FORWARD 設定爲IP_FORWARD=yes。

連線的目標主機可以看到您的路由器，但是並不知道您內部網路中產生封包的主機。這就是爲什麼這個技術稱爲偽裝。由於位址轉譯功能，所以路由器是任何回覆封包的第一個目的地。路由器必須識別這些內送封包並轉譯其目標位址，如此封包可轉遞到區域網路中的正確主機。

依據偽裝表格之內送交通的路由，是不可能從外面開啓對內部主機的連線。對於這類連線，表格中不會出現項目。此外，任何已經建立的連線在表格中都有指定給它的狀態項目，因此另一個連線無法使用此項目。

其結果是，您在一些應用程式通訊協定上可能會出現問題，如ICQ、cucme、IRC (DCC、CTCP) 和FTP (PORT 模式)。Netscape、標準的FTP 程式以及許多其他通訊協定使用PASV 模式。就封包過濾和偽裝而言，這種被動模式的問題較少。

34.1.3 防火牆基本原則

「防火牆」大概是使用最爲廣泛的字詞，用來描述提供及管理網路之間的連結機制，同時還能控制其間的資料流。嚴格來說，本節中描述的機制稱爲「封包過濾器」。封包過濾器根據特定標準(如通訊協定、通訊埠和IP 位址) 規範資料流。這樣允許您根據封包的位址，阻斷不應該送到您網路上的封包。例如，若要允許公用存取您的網頁伺服器，請明確開啓對應連接埠。不過，封包過濾器不會掃描具有正常位址的封包內容，如導向您網頁伺服器的那些封包。例如，即使內送封包意圖危害您網頁伺服器上的CGI 程式，封包過濾器仍會讓這些封包通過。

更有效但更複雜的機制是數種系統類型的組合，如與應用程式閘道或Proxy 互動的封包過濾器。這個時候，封包過濾器會拒絕預定給停用連接埠的任何封包。只有導向到應用程式閘道的封包才會被接受。此閘道或Proxy 會假裝是伺服器的實際用戶端。以這種意義而言，這類Proxy 會視爲通訊協定層級上應用程式所使用的偽裝主機。這種Proxy 的範例之一是Squid，一種HTTP Proxy 伺服器。若要使用Squid，瀏覽器必須設定成透過Proxy 進行通訊。要求的任何HTTP 頁面會從Proxy 快取取得，而快取中找不到的頁面會透過Proxy 從網際網路抓取。另一個範例則是，SUSE proxy-suite (proxy-suite) 爲FTP 通訊協定提供Proxy。

下節將焦點放在SUSE LINUX 隨附的封包過濾器。如需有關封包過濾及防火牆的詳細資訊，請參閱howto 套件中包含的「防火牆HOWTO」內容。如果安裝了此套件，請使用less /usr/share/doc/howto/en/txt/Firewall-HOWTO.gz 閱讀HOWTO。

34.1.4 SuSEfirewall2

SuSEfirewall2 是讀取 `/etc/sysconfig/SuSEfirewall2` 中變數集的程序檔，可以產生一組 iptables 規則。它定義三個安全性區域，但是以下範例組態僅考慮第一個和第二個區域：

外部區域 假設沒有方法可以控制外部網路上發生的情況，因此需要保護主機。在大部分的情況中，外部網路就是網際網路，但是也可能是其他不安全的網路，例如 WLAN。

內部區域 這是指私有網路，通常是指 LAN。如果此網路上的主機使用私有範圍的 IP 位址(請參閱節 22.1.2, “網路遮罩與路由”)，請啟用網路位址轉譯(NAT)，這樣內部網路上的主機即可存取外部網路。

廢除區域(DMZ) 儘管外部及內部網路可以連接位於此區域中的主機，但是這些主機本身無法存取內部網路。這種設定可以在內部網路前加上額外的防護線，因為 DMZ 系統與內部網路是隔離的。

過濾規則集未明確允許的任何網路流量類型會由 iptables 封鎖。因此，具有內送流量的每個介面必須放置在三個區域中的其中一個。對於每個區域，定義允許的服務或通訊協定。規則集僅套用到遠端主機產生的封包。本地產生的封包不會被防火牆攔截。

使用 YaST 可以執行組態(請參閱節 34.1.4, “使用 YaST 進行設定”)。也可以在檔案 `/etc/sysconfig/SuSEfirewall2` 中手動進行，該檔案的註解完整。不僅如此，在 `/usr/share/doc/packages/SuSEfirewall2/EXAMPLES` 中還提供一些範例案例。

使用 YaST 進行設定

自動防火牆組態

在安裝後，YaST 在所有已設定的介面上會自動啟動防火牆。如果已設定系統並在系統上啟動，YaST 會使用伺服器組態模組中的‘開啓防火牆中選取介面的連接埠’或‘開啓防火牆的連接埠’選項，以修改自動產生的防火牆組態。有些伺服器模組對話方塊具有‘防火牆詳細資料’按鈕功能，可以啟動其他服務和連接埠。YaST 防火牆組態模組僅能啓動、關閉或獨立重新設定防火牆。

YaST 的圖形組態對話方塊可從「YaST 控制中心」存取。選取‘安全性和使用者’→‘防火牆’。該組態一共分成七個部份，可以在畫面左手邊的樹狀結構直接存取。

啓動 在此對話方塊中設定啓動行爲。在預設的安裝中，SuSEfirewall2 已經在新安裝的系統上執行。您也可以在此啓動和停止防火牆。如果您想要測試自己目前的防火牆設定，請使用‘立即儲存設定並重新啓動防火牆’按鈕。

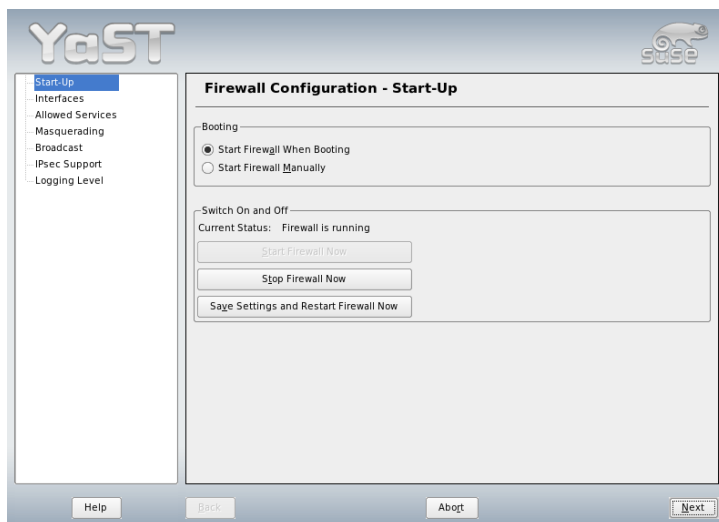


Figure 34.2: YaST 防火牆組態

介面 所有已知的網路介面都列在這裏。若要從區域移除介面，請選取介面，按‘變更’後選擇‘___no_zone___’。若要新增介面至區域，請選取介面，按‘變更’後選擇任何可用的區域。您也可以使用‘使用者定義’，以自己的設定建立特殊的介面。

允許的服務 您需要此選項以從系統提供服務至受保護的區域。根據預設，只有外部區域才會受到保護。在此例中，您需要讓服務能夠從外部主機明確看到。在‘選取區域所允許的服務’中選取適當的區域後，啓動個別的服務。

偽裝 偽裝允許您將內部網路隱藏，使網際網路這一類的外部網路看不到它。它也可以讓內部網路能輕易地存取外部網路。從外部網路發出對內部網路的

要求會遭到封鎖，但是從內部網路發出的要求，從外部看起來會像是從偽裝伺服器發出。

如果內部機器的特殊服務需要開放給外部網路使用，您可以針對個別服務增加特殊的重新指向規則。

廣播 在此對話方塊中，已設定允許廣播的UDP 埠。所需的埠號或服務必須加入個別的區域並以空白分隔。請參閱 `/etc/services` 檔案。

在這裏可以啓用不允許的廣播記錄。這有可能會有問題，因為Windows 主機使用廣播瞭解彼此，也因而產生許多不允許的封包。

IPsec 支援 您可以在此對話方塊中設定是否可以從外部網路允許IPsec 服務。信任封包的組態必須在‘詳細資料’之下設定。

記錄層次 記錄有兩種規則：允許和不允許的封包。允許為ACCEPTED，不允許為DROPPED 或REJECTED。您可以從‘記錄全部’、‘記錄關鍵’或‘全部不要記錄’來選取這兩個規則。

當完成防火牆組態時，請按‘下一步’結束此對話方塊。接著會顯示防火牆組態的區域導向摘要，在此您應該再次檢查所有的設定。所有已允許的服務，埠和通訊協定都會列在此摘要中。如果您想要回到組態，請使用‘上一步’，然後按‘接受’儲存組態。

手動組態

以下段落提供成功組態的逐步指示。每個組態項目會依據是否與防火牆或偽裝功能相關加以標示。這裡並沒有涵蓋組態檔中所提與DMZ(廢除區域)相關的觀點。這些觀點僅適用較大型組織的更複雜網路基礎結構(公司網路)，因為需要全面的組態以及對該主題的深入瞭解。

首先，使用YaST 模組Systems Services (RunLevel) 以啓用runlevel 中的SuSEfirewall2(最有可能是3 或5)。它在 `/etc/init.d/rc?.d/` 目錄中會設定SuSEfirewall2_* 程序檔的符號連結。

FW_DEV_EXT (防火牆、偽裝) 連結到網際網路的設備。如果是數據機連線，請輸入ppp0。如果是ISDN 連結，請使用ipp0。DSL 連線則使用dsl0。指定auto 使用與預設路由對應的介面。

FW_DEV_INT (防火牆、偽裝) 連結到內部私用網路的設備(如eth0)。如果沒有內部網路且防火牆僅保護執行的主機時，請留置空白。

FW_ROUTE (防火牆、偽裝) 如果需要偽裝功能，將此設定為yes。外界將無法看到您的內部主機，因為網際網路路由器會忽略其私用網路位址(例如，192.168.x.x)。

對於沒有偽裝的防火牆，如果想要允許存取內部網路，僅將此設定為yes。在這種情況下，您的內部主機需要使用正式註冊的IP。不過，通常您「不」應該允許外界存取您的內部網路。

FW_MASQUERADE (偽裝) 如果需要偽裝功能，將此設定為yes。這為內部主機提供幾乎直接的網際網路連線。在內部網路和網際網路的主機之間使用Proxy 伺服器較為安全。對於Proxy 伺服器所提供的服務並不需要偽裝。

FW_MASQ_NETS (偽裝) 指定要偽裝的主機或網路，在個別項目之間加上空格。例如：

```
FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
```

FW_PROTECT_FROM_INT (防火牆) 將此設定為yes，保護您的防火牆主機免於來自內部網路的攻擊。如果確實啓用，只有內部網路才可以使服務。另請參閱FW_SERVICES_INT_TCP 與FW_SERVICES_INT_UDP。

FW_SERVICES_EXT_TCP (防火牆) 輸入應該可用的TCP 連接埠。對於不應該提供任何服務的一般家用工作站，請留置空白。

FW_SERVICES_EXT_UDP (防火牆) 除非執行UDP 服務且希望讓外界使用，否則請留置空白。使用UDP 的服務包括DNS 伺服器、IPSec、TFTP、DHCP 以及其他。如果要讓外界使用，輸入要使用的UDP 連接埠。

FW_SERVICES_INT_TCP (防火牆) 利用此變數，定義可讓內部網路使用的服務。其表示法與FW_SERVICES_EXT_TCP 相同，但是設定是套用到「內部」網路。只有在FW_PROTECT_FROM_INT 設定為yes 時，才需要設定變數。

FW_SERVICES_INT_UDP (防火牆) 請參閱FW_SERVICES_INT_TCP。

設定防火牆後，測試您的設定。防火牆規則集的建立是以root 身份輸入SuSEfirewall2 start。接著使用telnet，例如，從外部主機查看是否已實際拒絕連線。之後，檢視/var/log/messages，在其中應該會看到如下內容：

```
Mar 15 13:21:38 linux kernel: SFW2-INext-DROP-DEFLT IN=eth0
OUT= MAC=00:80:c8:94:c3:e7:00:a0:c9:4d:27:56:08:00 SRC=192.168.10.0
DST=192.168.10.1 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=15330 DF PROTO=TCP
SPT=48091 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A061AFEB0000000001030300)
```

其他測試防火牆設定的套件為nmap或nessus。nmap的文件可在/usr/share/doc/packages/nmap中找到，而nessus的文件則是在安裝個別套件後的/usr/share/doc/packages/nessus-core目錄中。

34.1.5 若需詳細資訊

有關SuSEfirewall2套件的最新資訊及其他文件請參閱/usr/share/doc/packages/SuSEfirewall2.netfilter與iptables專案的首頁(<http://www.netfilter.org>)提供許多語言的大量文件。

34.2 SSH：安全網路作業

隨著有愈來愈多的電腦安裝在網路環境中，通常會需要從遠端位置存取主機。這通常是指使用者會傳送登入與密碼字串以供驗證。只要這些字串是以純文字傳送，它們就有可能被攔截和濫用以取得使用者帳戶的存取權，而授權的使用者卻完全未察覺。除了這將會把使用者的所有檔案暴露給攻擊者之外，不合法的帳戶有可能取得管理員或root存取權，或是滲透其他的系統。在過去，遠端連線是使用telnet建立，它並未提供對抗監聽加密形式或其他安全性機制的防護。有一些通訊通道是未防護的，像是傳統的FTP通訊協定以及某些遠端的複製程式。

SSH套裝軟體可以提供必要的保護，即加密驗證字串(通常是登入名稱與密碼)以及所有在主機間的其他資料交換。有了SSH，即使第三方記錄了資料流，但是內容已被加密，除非知道加密金鑰，否則無法將內容回復至純文字。因此SSH可以在非安全的網路上啓用安全通訊，例如網際網路。SUSE LINUX隨附的SSH為OpenSSH。

34.2.1 OpenSSH 套件

SUSE LINUX預設會安裝OpenSSH套件。ssh、scp以及sftp程式就變成了telnet、rlogin、rsh、rcp以及ftp的替代程式。在預設的組態中，只有使用OpenSSH公用程式且防火牆允許時，才能存取SUSE LINUX系統。

34.2.2 ssh 程式

使用ssh程式，就有可能登入遠端系統並互動式地工作。它會取代telnet與rlogin。slogin程式只是指向ssh的符號連結。例如，使用ssh sun指令登入sun主機。該主機接著會提示輸入sun的密碼。

在成功地驗證後，即可使用遠端指令行，或使用互動式應用程式，例如YaST。如果本地使用者名稱與遠端使用者名稱不同，您可以使用不同的登入名稱加上ssh -l augustine sun 或ssh augustine@sun。

此外，ssh 提供從遠端系統上執行指令(亦即rsh)的可能性。在下列範例中，在sun 主機上執行uptime 指令，並建立名稱爲tmp 的目錄。程式輸出會顯示在earth 主機的本地終端機上。

```
ssh otherplanet "uptime; mkdir tmp"
tux@otherplanet's password:
1:21pm up 2:17, 9 users, load average: 0.15, 0.04, 0.02
```

在這裏需要使用引號以使用一個指令傳送兩個指示。只有執行此動作，第二個指令才可以在sun 上執行。

34.2.3 scp — 安全複製

scp 會將檔案複製到遠端機器。它是rcp 的安全與加密的替代程式。例如，scp MyLetter.tex sun: 從earth 主機將MyLetter.tex 檔案複製到sun 主機。如果在earth 上的使用者名稱與sun 上的使用名稱不相同，請使用username@host 格式指定後者的使用者名稱。此指令並沒有-l 的選項。

在輸入正確的密碼後，scp 會啓動資料傳輸並顯示逐漸變長的星號列，以模擬進度列。除此之外，程式會永遠在進度列的右邊顯示到達的估計時間。指定-q 選項以抑制所有的輸出。

scp 也會爲整個目錄提供遞迴複製功能。scp -r src/ sun:backup/ 指令會複製src 目錄的整個內容(包括所有的子目錄)至sun 主機上的backup 目錄。如果這個子目錄不存在，就不會自動建立。

-p 選項會指示scp 保留未變更檔案的時間戳記。-c 可以壓縮資料傳輸。這可能會減少要傳輸的資料量，但是會造成處理器更重的負擔。

34.2.4 sftp — 安全檔案傳輸

可以使用sftp 程式取代scp，以進行安全的檔案傳輸。在sftp 工作階段期間，可以使用許多已知的ftp 指令。sftp 程式可能是比scp 更好的選擇，特別是在傳輸未知檔名的資料時。

34.2.5 SSH 精靈(sshd) — 伺服器端

若要使用SSH用戶端程式ssh與scp，伺服器(SSH精靈)必須在背景執行，監聽TCP/IP port 22的連線。精靈會在第一次啟動時產生三個金鑰組合。每個金鑰組合都是由私密與公用金鑰所組成。因此，此程序可稱為公用金鑰式。為了保證透過SSH通訊的安全性，必須限制只有系統管理員可以存取私密金鑰檔。預設安裝會據此設定檔案權限。SSH精靈只有在本地上需要私密金鑰，而且絕對不能將它給予任何人。公用金鑰元件(以.pub的副檔名來識別)會傳送至要求連線的用戶端。它們可讓所有的使用者讀取。

SSH用戶端所啓始的連線。等待SSH精靈與要求SSH用戶端交換識別資料，以比較通訊協定與軟體版本，並防止透過錯誤埠的連線。因為原始SSH精靈的子處理序會回覆要求，所以可以同時建立數個SSH連線。

至於SSH伺服器與SSH用戶端之間的通訊，OpenSSH可以支援SSH通訊協定的版本1與2。新安裝的SUSE LINUX系統預設值為版本2。若要在更新後繼續使用版本1，請遵照在/usr/share/doc/packages/openssh/README.SuSE中的指示。這個文件也描述如何只執行一些步驟，即將SSH1環境轉換成工作SSH2環境。

當使用SSH的版本1時，伺服器會傳送每小時由SSH精靈重新產生的公用主機金鑰與伺服器金鑰。這兩者都允許SSH用戶端加密自由選擇的工作階段金鑰，並且會將它傳送至SSH伺服器。SSH用戶端也會指示伺服器要使用的加密方法(密碼)。

SSH通訊協定的版本2並不需要伺服器金鑰。兩邊都是使用根據Diffie-Helman的演算法來交換金鑰。

私密主機與伺服器金鑰絕對需要解密工作階段的金鑰，而且不能從公用部份產生。只有連絡的SSH精靈可以使用私密金鑰解密工作階段金鑰(請參閱man/usr/share/doc/packages/openssh/RFC.nroff)。若要仔細監看此啓始連線階段，可以開啓SSH用戶端的-v詳細偵錯選項。

預設會使用SSH通訊協定的版本2。使用-l參數即可覆寫此預設以使用通訊協定的版本1。用戶端會在第一次與遠端主機連繫後，將所有的公用主機金鑰儲存在~/.ssh/known_hosts中。這將可防止外來的SSH伺服器進行「在中間攔截的攻擊」(外部SSH伺服器使用假冒的名稱與IP位址)。這類的攻擊通常都是由下列方式所偵測出來：在~/.ssh/known_hosts中未包含主機金鑰，或伺服器在缺少適當的私密對照組時無法解密工作階段的金鑰。

建議將儲存在/etc/ssh/中的私密與公用金鑰備份在安全的外部位置。以此方式，就可以偵測出金鑰的修改，並且可以在重新安裝後，再度使用舊的金鑰。這可讓使用者避免收到任何擾人的警告。儘管出現警告，但是如果已確認它確實是正確的SSH伺服器，就必須從~/.ssh/known_hosts移除關於此系統的現有項目。

34.2.6 SSH 驗證機制

目前以最簡單的形式所執行的實際驗證，是由上方所提及的輸入密碼所組成。SSH 的目標就是要引進安全且易於使用的軟體。因為它是用以取代 rsh 與 rlogin，SSH 也必須能夠提供適於平日使用的驗證方法。SSH 是藉由使用者所產生的另一個金鑰組合來完成此動作。SSH 套件為此提供了說明程式：ssh-keygen。在輸入 ssh-keygen -t rsa 或 ssh-keygen -t dsa 之後，就會產生金鑰組合，而且會提示您輸入儲存金鑰的基本檔案名稱。

確認預設值並回覆通關密語的要求。即使軟體建議空白的通關密語，一般建議輸入 10 至 30 個字元的文字以描述在此的程序。請勿使用簡短的文字或片語。重複通關密語以確認。接著，您將會看到儲存私密與公用金鑰的位置，在此例中，為 id_rsa 與 id_rsa.pub 檔案。

使用 ssh-keygen -p -t rsa 或 ssh-keygen -p -t dsa 以變更舊的通關密語。複製公用金鑰元件(在範例中為 id_rsa.pub) 至遠端機器，並將它儲存至 ~/.ssh/authorized_keys。下次建立連線時，將會要求您驗證自己的通關密語。如果沒有要求您驗證，請驗證這些檔案的位置與內容。

就長期而言，此程序比每次給密碼更為麻煩。因此，SSH 套件提供另一個工具：ssh-agent。此工具可保留了 X 工作階段期間的私密金鑰。整個 X 工作階段是啟動成 ssh-agent 的子處理序。執行這個的最簡單的方法就是在 .xsession 檔案的開頭將 usessh 變數設為 yes，並透過顯示管理員(例如 KDM 或 XDM) 登入。或者，輸入 ssh-agent startx。

現在您可以照常使用 ssh 或 scp。如果您已依照上方所述配送公用金鑰，就不會再提示您輸入密碼。請利用 xlock 之類的密碼保護應用程式，來終止或鎖定 X 工作階段。

在推出 SSH 通訊協定版本 2 後所造成的相關變更，也記載在 /usr/share/doc/packages/openssh/README.SuSE 檔案中。

34.2.7 X，驗證與轉寄機制

除了先前所述的安全性相關改善之外，SSH 也會簡化遠端 X 應用程式的使用。如果您以 -X 選項執行 ssh，就會自動在遠端機器上設定 DISPLAY 變數，而且所有的 X 輸出都會透過現有的 SSH 連線輸出至遠端機器。同時，未獲授權的個體將無法攔截以此方式檢視，並在遠端和本地上啟動的 X 應用程式。

透過新增 -A 選項，ssh-agent 驗證機制就會延續至下一部機器。以此方式，您不需輸入密碼就可以在不同的機器上工作，但僅限於您已配送公用金鑰至目的主機並將它適當地儲存在主機上。

兩種機制都是在預設值中停用，但是也可以隨時在整個系統的 /etc/ssh/sshd_config 組態檔中或使用者的 ~/.ssh/config 中啟動。

ssh 也可以用於重新導向TCP/IP 連線。在下列範例中，設定SSH 分別重新導向SMTP 與POP3 埠：

```
ssh -L 25:sun:25 earth
```

使用此指令後，任何導向`earth` 埠25 (SMTP) 的連線都會透過加密通道重新導向至sun 上的SMTP 埠。這對於那些使用SMTP 伺服器但卻沒有SMTP-AUTH 或POP-before-SMTP 功能的連線特別有用。從任何連線至網路的位置，都可以將電子郵件傳送至「主」郵件伺服器以進行傳遞。同樣地，在`earth` 上的所有POP3 要求(埠 110) 都可以使用此指令轉寄至sun 的POP3 埠：

```
ssh -L 110:sun:110 earth
```

兩個指令都必須以`root` 的身份執行，因為該連線必須以有權限的本地埠建立。電子郵件是由一般的使用者在現有的SSH 連線中傳送和擷取。SMTP 與POP3 主機都必須設成`localhost` 才能正常運作。上方所述的每個程式的其他資訊都可以在手冊頁面中找到，也可以在`/usr/share/doc/packages/openssh` 下的檔案中找到。

34.3 加密分割區和檔案

34.3.1 應用程式案例

每一位使用者都有一些機密資料，不想被第三方取得。連線越頻繁、越有機動性，處理資料時益發令人不放心。當第三方可以透過網路連線或直接實體存取時，為檔案或整個分割區加密，永遠是明智的。以下清單重點摘述一些可以想像的使用案例。

筆記型電腦 如果您旅行時攜帶筆記型電腦，為含有機密性資料硬碟的分割區加密是很棒的點子。在您的筆記型電腦弄丟或遭竊時，如果資料位於加密檔案系統或加密單一檔案，資料安全無虞。

抽取式媒體 USB 隨身碟或外接式硬碟和筆記型電腦一樣容易竊取。加密檔案系統提供防止第三方存取的保護。

34.3.2 使用YaST 設定密碼檔案系統

YaST 在安裝時提供檔案或分割區的加密與已經安裝的系統。因為加密檔案與現存分割區配置完全相合，所以任何時候都可建立加密檔案。要加密整個分割區，您需要固定分割區以便在分割區配置進行加密。依照預設，YaST 建議的標準分割提案，不包含加密分割區。在分割區的對話方塊手動新增。

在安裝時建立加密分割區

┌

密碼輸入

建立加密分割區的密碼時注意密碼安全性警告，並請將密碼牢記於心。若沒有密碼，不能存取加密資料。

└

詳述於節2.7.5, “磁碟分割”的YaST分割區精靈對話方塊，提供建議加密分割區需要選項。按一下‘建立’像是建立一般分割區時一樣。在開啓的對話方塊，輸入新分割區的分割參數，例如想要的格式化和裝置點。按一下‘加密檔案系統’完成建立步驟。在以下對話方塊設定密碼，接著，基於安全性理由請重複輸入密碼。按一下‘確定’，完成分割對話方塊後，就可以建立新的加密分割區。作業系統會在開機之際，掛載分割區之前，提示使用者輸入該密碼。

如果您不想在啓動時裝置加密分割區，提示輸入密碼後，請按一下(輸入)。接著拒絕再次輸入密碼。在此例中不會裝載加密檔案系統，而作業系統會繼續啓動，此為保護您資料安全的方式。一旦分割區裝載完成，所有使用者都可以使用它。

如果加密檔案系統只要在需要時才掛載，請核取‘fstab 選項’對話方塊的‘開機時不要掛載’。系統啓動時不會掛載對應分割區。要讓它之後能夠使用，請使用 `mount <name_of_partition> <mount_point>` 手動掛載。提示裝掛載分割區時，請輸入密碼。完成分割區後，使用 `umount name_of_partition` 取消裝載，以防止其他使用者存取分割區。

在執行系統上建立加密分割區

┌

在執行系統啓動系統加密

在執行系統上建立加密分割區，也可以使用與安裝期間相同的方式。然而，加密現存分割區會損毀它所有的資料。

└

在執行中的系統上，選取YaST控制中心的‘系統’→‘分割區’。按一下‘是’繼續。如上述，按一下‘編輯’，而不要選取‘建立’。其他程序則相同。

安裝加密檔案

就像使用分割區一樣，可以在保存機密性資料的單一檔案之中，建立加密檔案系統。從相同YaST對話方塊，建立加密檔案系統。選取‘加密檔案’，然後輸入要建

立的檔案路徑和檔案大小。接受格式化建議的設定值和檔案系統類型。然後指定裝載點，同時決定開機時是否裝載加密檔案系統。

加密檔案的優點是，無須變更硬碟分割就能進行新增加密檔案。他們透過迴路設備進行裝載，運作方式就跟一般分割區一樣。

使用vi 加密檔案

使用加密分割區的缺點是必須暫時裝載分割區，但是至少root 使用者可以存取資料。若要避免此狀況，可以在加密模式中使用vi。

使用vi -x filename 編輯新檔案。vi 會提示輸入密碼，然後加密檔案內容。每當您再次存取檔案，vi 會提示輸入正確密碼。

若確定要儲存，您甚至可以將加密文字檔置放於已有安全保護的分割區。這非常實用，因為已知vi 使用的加密機制不是非常嚴密的加密機制。

34.3.3 加密抽取式媒體內容

抽取式媒體，像是外接硬碟或USB 隨身碟，像其他硬碟一樣，無法被YaST 辨認。這些媒體上的檔案或分割區也可以依上述方法加密。開機時，不要選取裝載這些媒體，因為通常只有在系統運作時，才能連接它們。

34.4 安全性與機密性

Linux 或UNIX 系統的其中一個主要特性是能夠同時(多重使用者) 處理數個使用者，並允許這些使用者在同一部電腦上同時執行數個工作(多工作業)。再者作業系統具有透明化的網路。使用者通常不知道他們正在使用的資料與應用程式是由其本地機器提供或是透過網路所提供。

使用多個使用者的功能，不同使用者的資料必須分開儲存。必須保證安全性與隱私。即使在電腦可以透過網路連結前，資料安全性就已經是一個重要的議題。就像今日，最重要的問題就是在遺失資料或損毀資料的媒體上(大部份是指硬碟而言)，仍然能夠提供資料。

本小節主要是將焦點放在機密議題以及保護使用者隱私的方法上，但是它並未著墨太多關於全面安全性概念應該永遠包含的一些程序，即在適當的位置具有定期更新、可運作以及已測試的備份。如果没有這些，您可能很難將資料還原—不只是在某些硬體有缺陷的例子中，如果懷疑有人取得未授權的存取權並損毀檔案時也是如此。

34.4.1 本地安全性與網路安全性

存取資料有數種方式：

- 與具有所需資訊或具有電腦資料存取權的人員進行個人通訊
- 直接從電腦的主控台(實體存取)
- 透過序列線
- 使用網路連結

在所有的這些例子中，在使用者存取資源或是有問題的資料前，應該先驗證其身份。網頁伺服器在這方面可能限制比較少，但是您仍然不應該對任何網友揭露所有的個人資料。

在上方的清單中，第一個例子是需要與人頻繁的互動，例如，當您正在與銀行人員聯繫時，需要證明您就是擁有該銀行帳戶的人員。接著會要求您提供簽名、PIN 或密碼以證明您就是您所聲稱的那個人。在某些狀況下，有可能從某個知道的人問出一些資訊，只要使用一些巧妙的措詞提及一些已知的片斷資訊，就有可能贏得該人員的信賴。這個受害者有可能被引導逐漸揭露更多的資訊，但卻不自知。在駭客之間，稱此為「社會工程」。您只能以教育人們的方式以及以謹慎的方式來處理語言與資訊才能對抗這樣的受害。在攻擊者侵入電腦系統前，他們通常會嘗試鎖定接待人員、公司的客服人員或甚至是家庭成員。在許多狀況下，利用社會工程的攻擊通常都發現得很晚。

一個想要取得資料未授權存取權的人，也有可能使用傳統的方式，嘗試直接取得您的硬體。因此，應該善加保護機器，以防他人移動、替換或破壞其元件。這也適用於備份，甚至是任何的網路纜線或電線。另外也需要防護開機程序，因為有些眾所周知的按鍵組合，可能會引起不正常的行為。請設定 BIOS 與開機載入程式的密碼，以避免自己的電腦發生此狀況。

連接至序列埠的序列終端機仍然用於許多地方。不像網路介面，它們並不依賴網路通訊協定和主機通訊。它們使用簡單的纜線或紅外線埠，在設備之間來回傳送純文字字元。纜線本身是這類系統最弱的點：只要以舊型的印表機連接到系統，就可以輕易地記錄纜線上所執行的任何內容。印表機可以達成的動作也可以由其他方式來完成，端視攻擊所進行的動作而定。

讀取某個主機本地上的檔案所需的存取規則，就是在不同的主機上開啓與伺服器的網路連線。本地安全性與網路安全性之間是有區別的。這條界線在於資料必須放入封包以傳送到別的地方。

本地安全性

本地安全性是從電腦執行位置的實體環境開始。在符合您的期望與需求的適當位置安裝機器。本地安全性的主要目標就是要將使用者彼此區隔開來，如此就不會有其他的使用者可以擅用另一個使用者的權限或身份。這是一般所遵守的規則，但是對於擁有系統最高權限的root使用者更是特別需要遵守。因為root可以使用任何本地使用者的身份，而且系統不會提示它輸入密碼，就可以讀取任何本地儲存的檔案。

密碼

在Linux系統上，密碼當然不會以純文字儲存，而且輸入的文字字串也不會與儲存的型式直接符合。如果是這樣，只要有人取得該對應檔案的存取權，系統上的所有帳戶都將受到侵害。實際上，會加密儲存的密碼，而且每次輸入它時，還會再加密一次，然後再比較這兩個加密的字串。只要加密的密碼無法反轉計算成原始的文字字串，就不允許登入，這提供了更多的安全性。

實際上，這是由一種特殊的演算法所達成，也稱為「暗門演算法」，因為它只能往一個方向運作。取得加密字串的攻擊者將無法以直接再次套用相同的演算法來取得密碼。因此，在找到一個組合與加密的密碼相似前，它將需要測試所有可能的字元組合。使用8個字元長的密碼，將需要計算相當多的可能組合。

在70年代，有人主張此方法比其他方法更為安全，因為使用了相當慢的演算法，它需要好幾秒才能加密一個密碼。然而，在另一方面，PC已變得非常強大，可以在每秒內進行數十萬或甚至數百萬的加密。因此，加密的密碼不應該對一般的使用者顯示（一般使用者無法讀取/etc/shadow）。更重要的一點是密碼必須不易猜測，以防止密碼因為某些錯誤而變成可見。因此，將密碼如“tantalise”「翻譯」成“t@nt@lls3”並沒有太大的用處。

以某些看起來與字母相似的數字取代一個字的某些字母並不夠安全。使用字典猜測的密碼破解程式也可以猜測到這樣的替換。最好的方法是使用一個沒有一般意義但只對您個人有意義的文字，像是某句子或某書名文字的第一個字母，例如“The Name of the Rose” by Umberto Eco。這將可以給予下列安全的密碼：“TNotRbUE9”。相反的，像是“beerbuddy”或“jasmine76”，將很容易被對您只有一些瞭解的人士猜到。

開機程序

設定系統使其無法從軟碟或從CD開機，可以將設備整個移除或是設定BIOS的密碼，並將BIOS設成只允許從硬碟開機。一般而言，Linux系統是由開機載入程式所啟動，允許您將額外的選項傳遞至開機核心。在/boot/grub/menu.lst中設定額外的密碼以防止其他人在開機時使用像是這樣的參數（請參閱章8，開機載入器）。這對於系統的安全性非常重要。不只是核心本身是以root權限執行，它也是第一個授權者可以在系統啟動時授與root權限。

檔案許可權

一般而言，儘可能以限制最多的權限來進行指定的工作。例如，閱讀或撰寫電子郵件絕對不需要使用root的身份。如果郵件程式有問題，而這個問題有可能造成在程式啓動時，利用它以相同的程式權限來進行攻擊。遵照上方原則，可以減少可能的損毀。

在SUSE版本中所包含的200,000個以上的檔案權限是經過小心選擇的。安裝其他軟體或其他檔案的系統管理員應該極爲注意何時會這麼做，特別是在設定權限時。資深以及對安全性敏感的系統管理員，永遠會使用-l選項加上ls指令以取得龐大的檔案清單，這可允許它們立即偵測是否有任何不正確的檔案權限。不正確的檔案屬性並不只是代表檔案已經過變更或遭到刪除。這些修改過的檔案有可能是root曾經執行過它們，或是在組態檔中，程式可能以root的權限使用過這些檔案。這大幅地增加了攻擊者的機會。像這一類的攻擊稱爲布穀鳥式借巢孵蛋，因爲不同的使用者(鳥)執行(孵化)程式(蛋)，就像布穀鳥誘使其他的鳥孵化牠自己的蛋。

SUSE LINUX系統包含permissions、permissions.easy、permissions.secure以及permissions.paranoid檔案，全部都在/etc目錄中。這些檔案的目的就是用以定義特殊的權限，像是全球都可以寫入的目錄或是供檔案使用的Setuser ID位元(具有Setuser ID位元集的程式並不是以啓動它的使用者權限執行，而是以檔案擁有者的權限執行，大部份是指root而言)。管理員可以使用/etc/permissions.local檔案以新增他自己的設定值。

若要定義SUSE的組態程式使用上方哪些檔案以據此設定權限，請選取YaST中的‘安全性’。若要學習更多關於此主題的詳細資訊，請參閱/etc/permissions中的備註或是chmod的手冊頁(man chmod)。

緩衝區溢位與格式字串問題

每當程式應該處理使用者可以或可能變更的資料時，就應該特別謹慎處理，不過這個問題大部份是針對應用程式設計人員而言，而非一般使用者。程式設計人員務必確保其應用程式以正確的方式解譯資料，而不需將它們寫入空間不足以儲存它們的記憶體區域中。另外，程式應該使用爲該目的所定義的介面，以一致性的方式傳遞資料。

如果在寫入緩衝區時，未考量記憶體緩衝區的實際大小，就有可能發生「緩衝區溢位」。在某些例子中，此資料(由使用者所產生)使用了比緩衝區可用空間更多的空間。結果，所寫入的資料超過了緩衝區資料的結尾，因此在某些狀況下，儘可能讓程式執行使用者(而非程式設計人員)所設定的程式順序，而不只是處理使用者資料而已。這一類的問題有可能造成嚴重的後果，特別是如果使用特別的權限執行程式的話(請參閱節34.4.1, “檔案許可權”)。

格式字串問題的運作方式有些不同，但是它又是有可能導致程式出狀況的使用者輸入。大部份而言，是利用這些程式的錯誤以及特殊的權限(Setuid以及Setgid

程式)，這也表示您可以移除對應的執行權限，以保護資料與系統免於這類的問題。此外，最好的方法是使用最低可能的權限來套用原則(請參閱節34.4.1, “檔案許可權”)。

假使緩衝區溢出與格式字串問題是與使用者資料處理相關的問題，如果已指定存取權給本地帳戶，就不只可以利用它們。許多已報告的問題都有可能透過網路連結被利用。因此，應該將緩衝區溢位與格式字串問題，分類成與本地安全性以及與網路安全性相關。

病毒

與某些人的說法相反，有些病毒是在Linux上執行。然而，那些作者所發行的已知病毒是一種「概念證明」，以證明該技術如預期運作。目前為止沒有病毒已經蔓延。

如果没有寄主得以寄生，病毒將無法生存和蔓延。在我們的例子中，宿主將會是一個程式或是系統的重要儲存區域，例如主要開機記錄，也就是病毒的程式碼可以寫入的。由於多個使用者的功能，Linux可以將存取權限制為某些檔案，特別是某些重要的系統檔案。因此，如果您以root權限進行一般工作，將會增加系統受到病毒感染的機會。相反的，如果您遵照上方所提及的原則，儘可能使用最低權限，則感染病毒的機率便微乎其微。

除此之外，您絕對不要從網際網路上某些您不了解的網站執行程式。SUSE的RPM套件帶有加密的簽名以做為數位標籤，必須小心謹慎才能建立它們。病毒通常代表管理員或使用者缺乏必要的安全性認知，造成原先應該非常安全的系統有感染的風險。

病毒不應該與蠕蟲混淆，它完全屬於網路世界。蠕蟲並不需要宿主就可以散播。

網路安全性

網路安全性對於防護從外面開始的攻擊非常重要。需要使用者名稱與密碼以驗證使用者的一般登入程序，仍然是本地的安全性問題。在透過網路登入的特定例子中，區別兩個安全性領域。一直到實際驗證所發生的事，都是網路安全性，而之後所發生的任何事則為本地安全性。

X視窗系統以及X驗證

如一開始所提及，網路通透性是UNIX系統的其中一個主要特性。X是UNIX作業系統的視窗系統，可以印象深刻的方式來利用此功能。使用X基本上就可以輕易地在遠端主機中登入並啟動圖形程式，接著就可以透過網路在您的電腦上顯示該程式。

應該使用X伺服器以遠端顯示X用戶端時，X伺服器應該從未授權的存取來保護它(即該顯示)所管理的資源。更具體而言，某些權限必須指定給用戶端程式。使用X Window系統，就可以使用兩種方法來進行此動作，分別為以主機為基礎的存取控制以及以Cookie為基礎的存取控制。第一個是依賴用戶端應該執行的主機IP位址。負責控制的程式為xhost。xhost會將合法用戶端的IP位址輸入屬於X伺服器的小資料庫。然而，依賴IP位址的驗證並不安全。例如，如果有第二個使用者在傳送用戶端程式的主機上工作，則該使用者也可以存取X伺服器—就像有些人偷竊IP位址一樣。由於這些缺點，在此將不對此驗證方法做詳細的說明，但是您可以使用man xhost來學習它。

在以Cookie為基礎的存取控制例子中，所產生的字元字串只能讓X伺服器以及合法的使用者知道，就像某種ID卡。此Cookie(這個字並不是指一般的餅乾，而是指包含警語的中國幸運籤餅)是儲存在使用者主目錄的.Xauthority檔案中，並且提供給任何需要使用X伺服器的X用戶端來顯示視窗。使用者可以使用xauth工具來檢查.Xauthority檔案。如果您要重新命名.Xauthority或不小心從主目錄刪除檔案，您將無法開啓任何新的視窗或X用戶端。在Xsecurity的man頁面中，可以參閱更多關於X Window系統安全性機制的詳細資訊(man Xsecurity)。

SSH(安全外圍程序)可用以完整地加密網路連線，並將它以背景作業方式轉遞至X伺服器，但是使用者不會察覺到該加密機制。這又可稱為X轉遞。X轉遞是藉由模擬伺服器端的X伺服器並在遠端主機上設定外圍程序的DISPLAY變數來達成。您可以在節34.2, "SSH:安全網路作業"找到有關SSH的詳細資訊。

┌

如果您未考慮您所登入的主機是否為安全的主機，請勿使用X轉遞。啓用X轉遞後，攻擊者可以透過SSH連線進行驗證，以入侵您的X伺服器並查看您的鍵盤輸入。

└

緩衝區溢位與格式字串問題

如節34.4.1, "緩衝區溢位與格式字串問題"中所討論，應該將緩衝區溢位與格式字串問題，分類成與本地與網路安全性相關的問題。隨著這類問題的本地變化、網路程式的緩衝溢位，當成功地利用這些瑕疵時，大部份都會使用它們來取得root權限。即使不是這樣，攻擊者也可能使用該問題來取得未授權的本地帳戶，以利用其他可能存在於系統上的弱點。

透過網路連結以利用緩衝區溢位與格式字串問題，絕對是遠端攻擊最常見的形式。對於這些程式的不當利用以開拓新發現的安全性漏洞，通常會張貼在安全性郵件清單上。使用它們就可以鎖定弱點，而不需知道程式碼的詳細資訊。經過這

些年來，經驗顯示提供這些不當利用的程式碼對於更加安全的作業系統貢獻良多，很明顯地這是因為作業系統設計者被迫必須修正其軟體中的問題。有了免費軟體，每個人都可以存取原始程式碼(SUSE LINUX 是隨附在所有可用的原始程式碼)，而且任何找到弱點及其不當利用程式碼的人，都可以提交修補程式以修正對應的問題。

DoS — 阻絕服務攻擊法

此種攻擊的主要目的就是要封鎖伺服器程式或甚至是整個系統，有可能以數種方式達成：使伺服器超出負載、以垃圾封包使其忙碌，或利用遠端緩衝區溢位。通常DoS 攻擊的唯一目的就是要使服務消失。然而，一旦特定的服務變成無法使用時，通訊就有可能變成「攔截式攻擊」(封包攔截、TCP 連線攔截、偽裝式攻擊)以及DNS 定址攻擊的弱點。

攔截式：封包攔截、攔截、偽裝式攻擊

一般而言，由攻擊者所執行的遠端攻擊會將自己放在通訊主機之間，可稱為「攔截式攻擊」。幾乎所有類型的攔截式攻擊共同點就是，受害者通常不會察覺正在進行的事情。這類的攻擊有許多可能的變化，例如，攻擊者有可能接收某個連線要求並將它轉遞給目標機器本身。現在受害者已無意識地建立與錯誤主機的連線，因為另一端假裝自己是合法的目標機器。

攔截式攻擊最簡單的形式稱為「嗅探器」— 攻擊者「只是」監聽經過的網路流量。至於更複雜的攻擊，「攔截式攻擊」有可能嘗試接收已建立的連線(攔截)。如果要這麼做，攻擊者將需要一些時間分析封包，才能夠預測屬於該連線的TCP 序號。當攻擊者最後奪取目標主機的角色時，受害者將會注意到此，因為他們會取得錯誤訊息說明連線因失敗而終止。

有些通訊協定並沒有透過加密以防止攔截，只有在建立連線時執行簡單的驗證程序，使其更容易成為攻擊者攻擊的弱點。

「偽裝式攻擊」是一種將封包加以修改以包含假的來源資料的攻擊，通常是IP 位址。大部份主動攻擊的形式是依賴傳送這樣的假封包— 這在Linux 機器上只能由進階使用者執行(`root`)。

許多所提及的攻擊都是結合DoS 來執行。如果攻擊者見到有機會可以將某個主機出其不意地擊倒，即使只是很短的時間，這將使其更易於做主動的攻擊，因為該主機將有一段時間無法妨礙其攻擊。

DNS 定址攻擊

DNS 定址攻擊是指攻擊者破壞DNS 伺服器的快取，即使用偽裝的DNS 回覆封包以回覆它、嘗試使伺服器傳送某種資料給正在要求伺服器資訊的受害者。許多伺

服务器以IP位址或主机名称来维持与其他主机的信任关系。攻击者必须非常瞭解对主机之间信任关系的实际结构，以便将自己伪装成其中一个信任的主机。通常，攻击者会分析从服务器所收到的一些封包以取得所需的资讯。攻击者通常也需要在名称服务器锁定适时的DoS攻击。请使用加密连线，以便能够验证要连线的主机身份来保护自己。

蠕虫

一般常会将蠕虫与病毒混淆，但是在两者之间其实有一个明确的界限。不像病毒，蠕虫并不需要感染某个宿主程式才能存活。它们相当擅长于在网路结构上儘可能迅速散播。过去所出现的蠕虫(例如Ramen、Lion或Adore)都是利用服务器程式已知的安全性漏洞像是bind8或lprNG。防止蠕虫的入侵其实相当容易。假设在安全漏洞的发现以及蠕虫攻击服务器之间有些间隔，这样正是感染程式的更新版本準时提供的好机会。不过这只对管理员在有问题的系统上真正安装了安全性更新才有用。

34.4.2 一些一般的安全性秘訣與技巧

为了能妥善处理安全性问题，必须随时关注新开发，并瞭解最新的安全性问题。保护系统以防发生各种问题的最好方法，就是儘快取得和安装安全性公告所建议的更新套件。SUSE安全性公告是以邮件清单的方式公布，您可以至下列<http://www.novell.com/linux/security/securitysupport.html>连结来订阅。suse-security-announce@suse.de清单是第一手关于更新套件的资讯来源，并且包含SUSE主动贡献者的安全性团队成员。

邮件清单suse-security@suse.de是讨论任何所关心的安全性问题的好地方。您可在如上方所提的suse-security-announce@suse.de URL下订阅它。

bugtraq@securityfocus.com是全球其中一个最知名的安全性邮件清单。建议阅读这个清单，它每天会收到15到20个张贴。在<http://www.securityfocus.com>可以找到更多的详细资讯。

下列是关于处理基本安全性问题的有用规则清单：

- 根据每个工作儘可能都使用限制最多的权限集合原则，避免以root的身份执行一般工作。这将可减少得到布穀鳥式借巢孵蛋或病毒的風險，並防止自己犯錯。
- 如果有可能，請永遠嘗試使用加密的連線在遠端機器上工作。使用ssh(安全外圍程式)以取代telnet、ftp、rsh以及rlogin應該為標準的慣例。

- 避免使用僅依據IP 位址的驗證方法。
- 請試著將最重要的網路相關套件保持在最新的狀態，並訂閱對應的郵件清單以接收這類程式新版的公告(bind、sendmail、ssh 等等)。同樣的原則也適用於與本地安全性相關的軟體。
- 變更/etc/permissions 檔案以最佳化對系統安全性很重要的檔案權限。如果您從程式移除Setuid 位元，它有可能再也無法依照所需的方式執行工作。另一方面，在大部份的情況下，請想想程式也將不再有潛在性的安全性風險。您可以採取全球可以寫入的目錄與檔案的相似方式。
- 停用所有非必要的網路服務，讓伺服器正常運作。這可讓您的系統較為安全。開埠及LISTEN 插槽狀態，都可以在netstat 程式中找到。至於其選項，建議使用netstat -ap 或netstat -anp。-p 選項允許您查看哪些程序正在佔據哪個名稱下的埠。
比較netstat 的結果以及從主機外面對埠所做全面掃描的結果。進行此工作的最佳程式為nmap，它不只檢查機器的埠，也會針對哪些服務在等待這些埠做出摘要。然而，埠掃描有可能被視為一種侵略行為，因此若無管理員的明確允許，請勿在主機上執行此動作。最後，請記得不只要掃描TCP 埠，另外也必須掃描UDP 埠(-sS 與-sU 選項)。
- 如果要以可靠的方式來監督系統檔案的完整性，請使用tripwire 程式，可在SUSE LINUX 版本上找到。加密tripwire 所建立的資料庫以防有人篡改它。另外，在您的機器之外複製一份資料庫備份，將它儲存在未連接網路的外部資料媒體上。
- 在安裝協力廠商軟體時請謹慎小心。曾經有駭客將木馬程式建入安全軟體套件的tar 歸檔中，所幸很快就被發現了。如果您要安裝二進位的套件，請確定您對所下載的網站沒有疑慮。
SUSE 的RPM 套件是以GPG 簽名。SUSE 用以簽名的金鑰為：
ID:9C800ACA 2000-10-19 SUSE Package Signing Key ;build@suse.de;
Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA
rpm --checksig package.rpm 指令顯示解除安裝的套件，其檢查總數與簽名是否正確。尋找該版本第一張CD 上的金鑰以及全球大部份金鑰伺服器上的金鑰。
- 定期檢查使用者與系統檔案的備份。請考慮如果您未測試備份是否可以使用，它有可能沒有用處。
- 檢查記錄檔。請儘可能隨時撰寫小型的程序檔以搜尋可疑的項目。不可否認的，這不完全是一個瑣碎的工作。最後只有您才知道哪些項目是不尋常的，哪些才是正常的。

- 使用tcp_wrapper以限制存取在機器上所執行的個別服務，因此您對於哪個IP位址可以連線至服務具有明確的控制權。如需關於tcp_wrapper的進一步資訊，請參閱tcpd與hosts_access (man 8 tcpd、man hosts_access) 的手冊頁面。
- 使用SuSEfirewall以加強tcpd所提供的安全性(tcp_wrapper)。
- 將您的安全性措施設計成重複型：出現兩次的訊息總比沒有訊息來的好。

34.4.3 使用集中式安全性報告位址

如果您發現安全性相關的問題(請先檢查可用的更新套件)，請撰寫電子郵件給security@suse.de。請包含問題的詳細描述以及該套件的版本號碼。SUSE將會儘快嘗試傳送回覆。您最好能以PGP加密電子郵件。SUSE的PGP金鑰為：

```
ID:3D25D3D9 1999-03-06 SUSE Security Team <security@suse.de>  
Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5
```

這個金鑰也可以從<http://www.novell.com/linux/security/securitysupport.html>下載。

Linux 存取控制清單

本章節簡略介紹Linux 檔案系統的POSIX ACL (存取控制清單) 相關背景和功
能。ACL 可以看做是傳統檔案系統物件許可權概念的擴充。使用ACL，定義許
可權比傳統許可權概念更有彈性。

35.1	ACL 的優點	528
35.2	定義	528
35.3	處理ACL	529
35.4	應用程式的ACL 支援	536
35.5	若需詳細資訊	536

POSIX ACL 這個詞彙表示這是真正的POSIX (可攜式作業系統介面) 標準。各自的草稿標準POSIX 1003.1e 和POSIX 1003.2c 已經因多種因素而撤銷了。然而，在屬於UNIX 家族中很多系統發現的ACL，是以這些草稿為基礎，而且本章節敘述的系統ACL 實行，也是依循這兩個標準。要查閱它們，請到<http://wt.xpilot.org/publications/posix.1e/>。

35.1 ACL 的優點

傳統上，會為Linux 系統的每一個檔案物件，定義三組的許可權。這三組包括讀取(r)、寫入(w) 以及執行(x) 許可權，用於三種使用者類型(檔案擁有者、群組和其他使用者) 的每一個。此外，也可以設定設定使用者id、設定群組id 以及黏貼位元。這個小小的概念對於大部分實際情況已經足夠了。不過，對於更複雜的案例或進階應用程式，系統管理員以前必須使用一些訣竅才能克服傳統許可權概念的限制。

ACL 可以用於需要擴充傳統檔案許可權概念的狀況。它們允許指定許可權給個別使用者或群組，即使這些許可權並不對應至原始擁有者或所屬群組。存取控制清單是Linux 核心的特性之一，目前ReiserFS、Ext2、Ext3、JFS 和XFS 都能支援。使用ACL 可以理解複雜的情況，無需在應用程式層次上實行複雜的許可權模型。

以Linux 伺服器取代Windows 伺服器時，ACL 的優點特別明顯。部份連線的工作站還可以在移轉之後，繼續在Windows 下執行。Linux 系統提供檔案和列印服務給含Samba 的Windows 用戶端使用。假設Samba 支援存取控制清單，使用者許可權可以同時在Linux 伺服器和使用圖形使用者介面的Windows (限Windows NT 和更新的版本) 上設定。透過winbindd，還可以指定許可權給只存取Windows 領域中而且沒有Linux 伺服器任何帳戶的使用者。

35.2 定義

使用者類別 傳統POSIX 許可權概念使用三種使用者類別，用於在檔案系統中指定許可權：擁有者、所屬群組以及其他使用者。三種許可權位元可以設定給每一個使用者類別，提供許可權來讀取(r)、寫入(w) 以及執行(x)。

存取ACL 所有檔案系統物件(檔案和目錄) 的使用者和群組存取許可權，是透過存取ACL 的方法來決定。

預設ACL 預設的ACL 只能套用至目錄。它們在檔案系統物件建立時，會決定從其上層目錄繼承的許可權。

ACL 項目 每一個ACL包含一組的ACL項目。ACL項目包含類型(請參閱表格35.1, “ACL項目類型”), 使用者的識別符號或項目參照的群組, 以及一組許可權。對於部份項目類型, 群組或使用者的識別字尚未定義。

35.3 處理ACL

表格35.1, “ACL項目類型” 摘錄6種可能的ACL項目類型, 每個類型定義一個使用者或一群組使用者的許可權。**擁有者**項目定義使用者所屬檔案或目錄的許可權。**所屬群組**項目定義檔案的所屬群組許可權。超級使用者可以使用chown或chgrp變更擁有者或所屬群組, 在這樣情況下, 擁有群組項目會參照新擁有者和所屬群組。每個**命名的使用者**項目定義使用者許可權, 該項目指定於項目的辨識符號欄位, 此欄位在表格35.1, “ACL項目類型” 中是以文字形式顯示在中央欄位。每個**命名的群組**項目定義群組許可權, 該許可權指定於項目的辨識符號欄位。只有命名的使用者和命名的群組擁有的辨識符號欄位不是空的。其他項目定義所有其他使用者許可權。

透過定義那些有效項目的許可項目以及遮罩的項目, 遮罩項目會進一步限制由**命名的使用者**、**命名的群組**以及**所屬群組**項目授與的許可權。如果許可權存在於之前提及的任一項目和遮罩中, 它們都是有效的。只有包含於遮罩或實際項目的許可權是無效的—表示未授與此許可權。定義在**使用者**和**所屬群組**項目的所有許可權永遠有效。表格35.2, “遮罩存取許可權” 中的範例會示範此機制。

有兩種基本類型的ACL: **最小值ACL**只包含**擁有者**、**所屬群組**和其他類型的項目, 對應檔案和目錄的傳統許可權位元。**延伸ACL**則不僅如此。它必須包含**遮罩**項目而且可以包含**命名使用者**和**命名群組**類型的多個項目。

Table 35.1: ACL 項目類型

類型	文字形式
擁有者	user::rwx
命名的使用者	user:name:rwx
所有群組	group::rwx
命名的群組	group:name:rwx
遮罩	mask::rwx
其他	other::rwx

Table 35.2: 遮罩存取許可權

項目類型	文字形式	許可權
命名的使用者	user:geeko:r-x	r-x
遮罩	mask::rw-	rw-
	有效許可權:	r--

35.3.1 ACL 項目和檔案模式許可權位元

圖形35.1, “最小ACL: ACL 項目比較許可權位元”和圖形35.2, “延伸ACL: ACL 項目比較許可權位元”列舉最小值ACL和延伸ACL的兩個案例。圖的結構分成三個區塊—左方區塊顯示ACL項目的類型格式, 中間區塊顯示範例ACL, 而右方區塊則顯示傳統許可權概念的許可權位元, 例如, 以ls -l顯示。在二者案例中, 擁有者類別許可權會對應至ACL項目擁有者。其他類別許可權對應個別ACL項目。不過, 群組類別許可權的對應與這兩個案例不同。

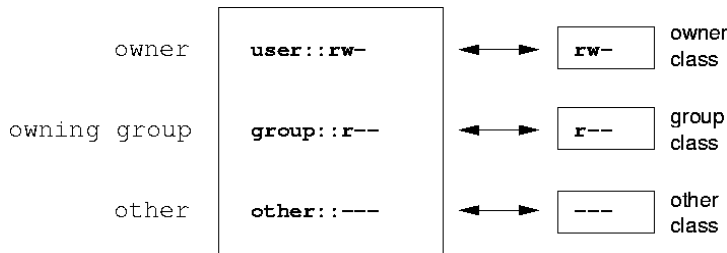


Figure 35.1: 最小ACL: ACL 項目比較許可權位元

在最小ACL案例中(不含遮罩), 群組類別許可權會對應至ACL項目所屬群組。這將顯示於圖形35.1, “最小ACL: ACL 項目比較許可權位元”。在延伸ACL案例中(含遮罩), 群組類別許可權是對應至遮罩項目。這會顯示於圖形35.2, “延伸ACL: ACL 項目比較許可權位元”。

此對應方法可以確定應用程式互動順暢, 無論它們是否擁有ACL支援。透過許可權位元方法所指定的存取許可權, 代表藉由ACL所做的所有其他「微調」上限。對許可權位元所做的變更會由ACL反映, 反之亦然。

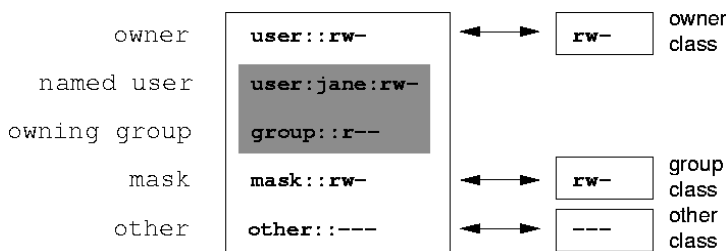


Figure 35.2: 延伸ACL：ACL項目比較許可權位元

35.3.2 含存取ACL的目錄

存取ACL的處理如以下範例所示範：

建立目錄之前，使用`umask`指令來定義每次建立檔案物件時，要遮罩的存取許可權。指令`umask 027`會指定完整範圍的許可權(0)給擁有者，拒絕群組寫入存取(2)以及不提供其他使用者任何許可權(7)，以設定預設許可權。`umask`實際上會遮罩相應的許可權位元或關閉它們。如需詳細資訊，請參閱相應的man頁面(`man umask`)。

`mkdir mydir`應該會以`umask`設定的預設許可權建立`mydir`目錄。使用`ls -dl mydir`檢查所有許可權是否指定正確。範例的輸出如下：

```
drwxr-x--- ... tux project3 ... mydir
```

使用`getfacl mydir`檢查ACL的啓始狀態。會提供類似以下的資訊：

```
# file: mydir
# owner: tux
# group: project3
user::rwx
group::r-x
other::---
```

`getfacl`的輸出會精確反應許可權位元以及ACL項目的對應，說明於節35.3.1，“ACL項目和檔案模式許可權位元”。前三個輸出行顯示目錄的名稱、擁有者和所屬群組。接下來三行包含三個ACL項目擁有者、所屬群組和其他。實際上，在此最小ACL案例中，`getfacl`指令不會產生任何您以`ls`指令無法取得的資訊。

以下列指令修改ACL以指定讀取、寫入和執行許可權給另一個使用者geeko以及另一個群組mascots：

```
setfacl -m user:geeko:rwx,group:mascots:rwx mydir
```

選項-m提示setfacl修改現有的ACL。以下引數指示要修改的ACL項目(多個項目由逗點分開)。最後部份指定這些修改要套用至什麼目錄名稱。使用getfacl指令，查看產生的ACL。

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
other:---
```

除了為使用者geeko和群組mascots實施的項目之外，已經產生一個遮罩項目。會自動設定遮罩項目讓所有的許可權生效。除非使用-n停用此功能，setfacl會將現有遮罩項目調整為修改過的設定值。遮罩會定義群組類別中所有項目的最大有效存取許可權。此包括命名使用者、命名群組以及所屬群組。ls -dl mydir顯示的類別群組許可權位元，對應至遮罩項目。

```
drwxrwx---+ ... tux project3 ... mydir
```

輸出的第一欄現在包含額外的+，指示此項目有一個延伸ACL。

根據ls指令的輸出，遮罩項目的許可權包括寫入存取。傳統上，此類的許可權位元表示所屬群組(在此為project3)也擁有目錄mydir的寫入存取。然而，所屬群組的有效存取許可權會對應至許可權的重疊部分，重疊部分定義在所屬群組和遮罩—在本範例中是r-x(請參閱表格35.2,“遮罩存取許可權”)。就範例中所屬群組的有效許可權而言，即使新增ACL項目之後也不會變更有效所有權。

使用setfacl或chmod，編輯遮罩項目。例如，使用chmod g-w mydir。接著ls -dl mydir顯示如下：

```
drwxr-x---+ ... tux project3 ... mydir
```

getfacl mydir提供下列輸出：

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx      # effective: r-x
group::r-x
group:mascots:rwx   # effective: r-x
mask::r-x
other::---
```

執行 `chmod` 指令，從群組類別位元移除寫入許可權之後，`ls` 指令的輸出就可以知道遮罩位元必須同時變更：寫入許可權再次限制為 `mydir` 的擁有者。`getfacl` 的輸出可確認此項結果。此輸出針對有效許可權未對應至原始許可權的所有項目包含一個註解，因為它們是根據遮罩項目過濾的。可以隨時使用 `chmod g+w mydir` 復原原始許可權。

35.3.3 含預設ACL 的目錄

目錄可以有一個預設的ACL，它是特別的ACL。建立物件時，它定義目錄中物件繼承的存取許可權。預設ACL 會影響子目錄和檔案。

預設ACL 的作用

子目錄的預設ACL，其許可權傳送至檔案以及子目錄的方法有兩種：

- 子目錄繼承上層目錄的預設ACL，當成預設ACL，也當成存取ACL。
- 檔案會將預設ACL 繼承為存取ACL。

建立檔案系統的所有系統呼叫，會使用 `mode` 參數，它會為新建立的檔案系統物件，定義存取許可權。如果上層目錄沒有預設ACL，`umask` 定義的許可權位元會從 `mode` 參數傳送的許可權中去除，將結果指定給新物件。如果上層目錄的預設ACL 存在，指定給新物件的許可權位元會對應至 `mode` 參數的許可權重疊部份，以及定義在預設ACL 的許可權。`umask` 在此案例中不予處理。

預設ACL 的應用

以下三個範例顯示目錄和預設ACL 的主要作業：

1. 新增預設ACL 至現有的目錄 `mydir`，透過：

```
setfacl -d -m group:mascots:r-x mydir
```

setfacl 指令的 -d 選項，提示 setfacl 在預設 ACL 執行以下修改 (-m 選項)。

仔細查看指令的結果：

```
getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
other:---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other:---
```

getfacl 傳回存取 ACL 和預設 ACL 二者。預設 ACL 是由開頭為 default 的所有行形成的。雖然您只是在預設 ACL 的 mascots 群組的一個項目執行 setfacl 指令，不過 setfacl 會自動從存取 ACL 複製所有其他項目來建立有效的預設 ACL。預設 ACL 對於存取許可權沒有立即的作用。它們只會在建立檔案系統物件時才有作用。這些新物件只會從其上層目錄的預設 ACL 繼承許可權。

2. 在下一個範例中，使用 mkdir，在 mydir 目錄中建立子目錄，它會繼承預設 ACL。

```
mkdir mydir/mysubdir
```

```
getfacl mydir/mysubdir
```

```
# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwx
group::r-x
group:mascots:r-x
```



```
mask::r-x
other::---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other::---
```

如預期般，新建立的子目錄mysubdir擁有上層目錄預設ACL的許可權。mysubdir的存取ACL與mydir的預設ACL完全相同。目錄傳給其從屬物件的預設ACL，也是如此。

3. 使用touch於mydir目錄建立檔案，例如，touch mydir/myfile。接著ls -l mydir/myfile會顯示如下：

```
-rw-r-----+ ... tux project3 ... mydir/myfile
```

getfacl mydir/myfile 輸出，如下：

```
# file: mydir/myfile
# owner: tux
# group: project3
user::rw-
group::r-x          # effective:r--
group:mascots:r-x   # effective:r--
mask::r--
other::---
```

建立新檔案時，touch會使用值為0666的模式。這表示檔案建立後，所有使用者類別都有讀取和寫入的許可權，而且在umask或預設ACL中沒有其它限制(請參閱節35.3.3,“預設ACL的作用”)。實際上，這表示所有未包含在mode值中的存取許可權，會從各自的ACL項目移除。雖然未從群組類別的ACL項目移除許可權，將遮罩項目修改為遮罩許可權，不在此模式中設定。

此方法可以確定應用程式(例如編譯器)與ACL的互動平順。您可以建立存取許可權受限的檔案，然後標示成可執行。mask機制可以保證正確的使用者和群組可以執行它們。

35.3.4 ACL 檢查演算法

任何程序或應用程式授與存取ACL保護的檔案系統物件前，會先套用檢查演算法。有個基本規則是，ACL項目會按照以下順序檢查：擁有者、命名使用者、

所屬群組或命名群組以及其他。處理的存取合乎最符合程序的項目。許可權不會累積。

如果程序屬於一個以上的群組而且可能符合多個群組項目，就會變得更複雜。項目是從符合必要許可權的合適項目中隨機選取的。它與觸發最後的結果“access granted”的項目無關。同樣地，如果適合的群組項目不包含需要的許可權，隨機選取的項目會觸發最後的結果「拒絕存取」。

35.4 應用程式的ACL 支援

ACL 可以用來實行很複雜的許可權情況，符合現在的應用程式需求。傳統許可權概念和ACL 可以使用聰明的方法結合。基本檔案指令(cp、mv、ls 等等) 都支援ACL，Samba 也支援ACL。

可惜的是，許多編輯器和檔案管理員都不支援ACL。例如，使用Konqueror 複製檔案時，這些檔案的ACL 會遺失。使用編輯器修改檔案時，檔案的ACL 有時候會保留，有時候則不保留，由編輯器的備份模式決定。如果編輯器將變更寫入原始檔案，會保留存取ACL。如果編輯器將更新的内容儲存至新檔案，後來重新命名成舊檔案名稱，則除非編輯器支援ACL，否則ACL 可能會遺失。除了star 歸檔設備之外，目前沒有備份應用程式可以保存ACL。

35.5 若需詳細資訊

如需關於ACL 的詳細資訊，請參閱<http://acl.bestbits.at/>。請參閱getfacl(1)、acl(5) 和setfacl(1) 的man 頁面。

系統監視公用程式

有許多程式和機制可用來檢查您的系統狀態，其中一些會在這裡介紹。另外還會介紹一些例行工作使用的公用程式，及其重要參數。

36.1	開啓檔案的清單：lsof	538
36.2	使用者存取檔案：fuser	539
36.3	檔案內容：stat	539
36.4	USB 設備：lsusb	540
36.5	SCSI 設備的相關資訊：scsiinfo	541
36.6	處理程序：top	541
36.7	程序清單：ps	542
36.8	程序樹：pstree	543
36.9	功能介紹：w	544
36.10	記憶體的使用：free	545
36.11	核心環狀緩衝區：dmesg	546
36.12	檔案系統及其使用：mount、df 和 du	546
36.13	/proc 檔案系統	547
36.14	vmstat、iostat 及 mpstat	549
36.15	procinfo	549
36.16	PCI 資源：lspci	550
36.17	執行程式的系統呼叫：strace	551
36.18	一個程式執行階段的程式庫呼叫：ltrace	552
36.19	指定需要的程式庫：ldd	552
36.20	其它有關ELF二進位的資訊	553
36.21	程序間的通訊：ipcs	554
36.22	使用time 進行時間管理	554

所有介紹的指令都會有相關的輸出範例。在這些範例中，第一行為指令本身(在金錢符號提示之後)。會以方括號[...] 表示備註，備註較長時會進行換行。因為備註過長而換行時，會以反斜線(\) 斷行。

```
$ command -x -y
output line 1
output line 2
output line 3 is annoyingly long, so long that \
    we have to break it
    output line 3
    [...]
    output line 98
output line 99
```

為了介紹更多公用程式，說明的部份保持簡短。如需所有指令的相關資訊，請參閱man 頁面。大部份的指令都能解讀參數--help，此參數可產生一份可能參數的簡短清單。

36.1 開啓檔案的清單：lsdf

如果要檢視為了程序而開啓的所有檔案之清單及其程序ID (PID)，可使用-p。例如，要檢視目前外圍程序正在使用的所有檔案，可輸入：

```
$ lsdf -p $$
COMMAND  PID USER  FD  TYPE DEVICE  SIZE  NODE NAME
zsh      4694  jj    cwd  DIR   0,18   144 25487368 /suse/jj/t (totan:/real-home/jj)
zsh      4694  jj    rtd  DIR   3,2    608 2 /
zsh      4694  jj    txt  REG   3,2   441296 20414 /bin/zsh
zsh      4694  jj    mem  REG   3,2  104484 10882 /lib/ld-2.3.3.so
zsh      4694  jj    mem  REG   3,2  11648 20610 /usr/lib/zsh/4.2.0/zsh/xlimits.s
o
[...]
zsh      4694  jj    mem  REG   3,2  13647 10891 /lib/libdl.so.2
zsh      4694  jj    mem  REG   3,2  88036 10894 /lib/libnsl.so.1
zsh      4694  jj    mem  REG   3,2 316410 147725 /lib/libncurses.so.5.4
zsh      4694  jj    mem  REG   3,2 170563 10909 /lib/tls/libm.so.6
zsh      4694  jj    mem  REG   3,2 1349081 10908 /lib/tls/libc.so.6
zsh      4694  jj    mem  REG   3,2 56 12410 /usr/lib/locale/de_DE.utf8/LC_TE
LEPHONE
[...]
zsh      4694  jj    mem  REG   3,2 59 14393 /usr/lib/locale/en_US/LC_NUMERIC
zsh      4694  jj    mem  REG   3,2 178476 14565 /usr/lib/locale/en_US/LC_CTYPE
zsh      4694  jj    mem  REG   3,2 56444 20598 /usr/lib/zsh/4.2.0/zsh/computil.
so
zsh      4694  jj    0u   CHR 136,48 50 /dev/pts/48
zsh      4694  jj    1u   CHR 136,48 50 /dev/pts/48
zsh      4694  jj    2u   CHR 136,48 50 /dev/pts/48
zsh      4694  jj    10u  CHR 136,48 50 /dev/pts/48
```

已使用特殊外圍程序變數\$\$，其值為外圍程序的程序ID。

指令 `lsdf` 列出所有目前開啓的檔案，使用時不需任何參數。因為通常其中有數千個檔案開啓，將這些檔案全部列出並不太有用。不過，可將這份包含所有檔案的清單結合搜尋功能，進而產生有用的清單。例如，將用到的字元設備全部列出：

```
$ lsdf | grep CHR
sshd      4685    root  mem   CHR   1,5    45833  /dev/zero
sshd      4685    root  mem   CHR   1,5    45833  /dev/zero
sshd      4693    jj    mem   CHR   1,5    45833  /dev/zero
sshd      4693    jj    mem   CHR   1,5    45833  /dev/zero
zsh       4694    jj    0u    CHR  136,48 50     /dev/pts/48
zsh       4694    jj    1u    CHR  136,48 50     /dev/pts/48
zsh       4694    jj    2u    CHR  136,48 50     /dev/pts/48
zsh       4694    jj    10u   CHR  136,48 50     /dev/pts/48
X         6476    root  mem   CHR   1,1    38042  /dev/mem
lsdf      13478   jj    0u    CHR  136,48 50     /dev/pts/48
lsdf      13478   jj    2u    CHR  136,48 50     /dev/pts/48
grep     13480   jj    1u    CHR  136,48 50     /dev/pts/48
grep     13480   jj    2u    CHR  136,48 50     /dev/pts/48
```

36.2 使用者存取檔案：fuser

這可用來判定哪些程序或使用者目前正存取著特定的檔案。例如，假設您要將裝載在 `/mnt` 上的檔案系統取消裝載，但 `umount` 指令傳回「設備忙碌」。接著可使用 `fuser` 指令來判定哪些程序正存取著此設備：

```
$ fuser -v /mnt/*

/mnt/notes.txt          USER          PID ACCESS COMMAND
                        jj            26597 f....  less
```

當另一個終端機上所執行的 `less` 程序結束時，即可成功地解除裝載檔案系統。

36.3 檔案內容：stat

指令 `stat` 會顯示檔案內容：

```
$ stat xml-doc.txt
  File: `xml-doc.txt'
  Size: 632          Blocks: 8          IO Block: 4096   regular file
Device: eh/14d Inode: 5938009 Links: 1
Access: (0644/-rw-r--r--)  Uid: (11994/      jj)   Gid: (   50/      suse)
Access: 2004-04-27 20:08:58.000000000 +0200
Modify: 2003-06-03 15:29:34.000000000 +0200
Change: 2003-07-23 17:48:27.000000000 +0200
```

參數`--filesystem`會產生特定檔案的檔案系統詳細內容：

```
$ stat . --filesystem
File: "."
  ID: 0          Namelen: 255      Type: ext2/ext3
Blocks: Total: 19347388  Free: 17831731  Available: 16848938  Size: 4096
Inodes: Total: 9830400  Free: 9663967
```

使用z 外圍程序(zsh)時，您必須輸入`/usr/bin/stat`，因為z 外圍程序有一個內建在stat 中的外圍程序，並且包含不同的選項以及不同的輸出格式：

```
% type stat
stat is a shell builtin
% stat .
device 769
inode 4554808
mode 16877
nlink 12
uid 11994
gid 50
rdev 0
size 4096
atime 1091536882
mtime 1091535740
ctime 1091535740
blksize 4096
blocks 8
link
```

36.4 USB 設備：lsusb

指令`lsusb`會列出所有USB 設備。加上選項`-v`，會列出更多細節清單。詳細資訊讀取自目錄`/proc/bus/usb/`。在USB 隨身碟連接上之後，下列是`lsusb`的輸出。最後一行表示有新的設備存在。

```
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 001 Device 018: ID 0402:5634 ALi Corp.
```

36.5 SCSI 設備的相關資訊：scsiinfo

指令 `scsiinfo` 會列出 SCSI 設備的資訊。加上選項 `-l`，可列出所有系統已知的 SCSI 設備(類似資訊經由指令 `lsscsi` 也可取得)。下列為 `scsiinfo -i /dev/sda` 的輸出，提供硬碟相關資訊。選項 `-a` 可提供更多資訊。

```
Inquiry command
-----
Relative Address          0
Wide bus 32              0
Wide bus 16              1
Synchronous neg.        1
Linked Commands          1
Command Queueing        1
SftRe                    0
Device Type              0
Peripheral Qualifier     0
Removable?              0
Device Type Modifier    0
ISO Version              0
ECMA Version             0
ANSI Version             3
AENC                    0
TrmIOP                   0
Response Data Format     2
Vendor:                  FUJITSU
Product:                 MAS3367NP
Revision level:         0104A0K7P43002BE
```

出現一個問題清單，內含兩個硬碟的錯誤區塊表格：第一個是由廠商所提供(廠商表格)，第二個是作業時出現的錯誤區塊清單(成長的表格)。若成長表格中的項目數量持續增加，此時最好更換硬碟。

36.6 處理程序：top

指令 `top` (即「程序表(table of processes)」的縮寫) 會顯示一個程序清單，每隔兩秒自動更新一次。要終止程式，請按 `Q`。在顯示一次程序清單之後，參數 `-n 1` 會終止程式。以下是 `top -n 1` 指令的輸出範例：

```
top - 14:19:53 up 62 days, 3:35, 14 users, load average: 0.01, 0.02, 0.00
Tasks: 102 total, 7 running, 93 sleeping, 0 stopped, 2 zombie
Cpu(s): 0.3% user, 0.1% system, 0.0% nice, 99.6% idle
Mem: 514736k total, 497232k used, 17504k free, 56024k buffers
Swap: 1794736k total, 104544k used, 1690192k free, 235872k cached
```

```

    PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  Command
  1426 root        15   0 116m  41m  18m  S   1.0   8.2   82:30.34 X
 20836 jj          15   0   820  820  612  R   1.0   0.2    0:00.03 top
     1 root        15   0   100   96   72  S   0.0   0.0    0:08.43 init
     2 root        15   0     0     0     0  S   0.0   0.0    0:04.96 keventd
     3 root        34  19     0     0     0  S   0.0   0.0    0:00.99 ksoftirqd_CPU0
     4 root        15   0     0     0     0  S   0.0   0.0    0:33.63 kswapd
     5 root        15   0     0     0     0  S   0.0   0.0    0:00.71 bdflush
    [...]
  1362 root        15   0   488  452  404  S   0.0   0.1    0:00.02 nscd
  1363 root        15   0   488  452  404  S   0.0   0.1    0:00.04 nscd
  1377 root        17   0    56   4     4  S   0.0   0.0    0:00.00 mingetty
  1379 root        18   0    56   4     4  S   0.0   0.0    0:00.01 mingetty
  1380 root        18   0    56   4     4  S   0.0   0.0    0:00.01 mingetty

```

如果您在top正在執行時按 \textcircled{F} ，會出現一個功能表，您可以利用它來變更輸出格式。

參數-U UID 只監視與特定使用者有關的程序。以使用者的使用者ID來取代 $\langle UID \rangle$ 。top -U \$(id -u username) 指令會傳回使用者UID，即使用者名稱，並顯示使用者的處理程序。

36.7 程序清單：ps

指令ps會產生一份程序清單。如果加入參數r，則只會顯示正在使用運算時間的程序。

```
$ ps r
  PID TTY          STAT TIME COMMAND
 22163 pts/7        R    0:01 -zsh
  3396 pts/3        R    0:03 emacs new-makedoc.txt
 20027 pts/7        R    0:25 emacs xml/common/utilities.xml
 20974 pts/7        R    0:01 emacs jj.xml
 27454 pts/7        R    0:00 ps r
```

寫入此參數時不可包含減號。寫入的參數有時會包含減號，有時則不包含。man 頁面很可能會嚇壞新的使用者，不過還好有ps --help 指令，它會產生一個簡短的說明頁面來提供協助。

如果要核對執行中的emacs 程序數目，請使用：


```
$ ps x | grep emacs
1288 ?          S          0:07 emacs
3396 pts/3      S          0:04 emacs new-makedoc.txt
3475 ?          S          0:03 emacs .Xresources
20027 pts/7      S          0:40 emacs xml/common/utilities.xml
20974 pts/7      S          0:02 emacs jj.xml
```

```
$ pidof emacs
20974 20027 3475 3396 1288
```

參數 `-p` 會利用程序ID 來選擇處理程序：

```
$ ps www -p $(pidof xterm)
  PID TTY          STAT       TIME COMMAND
  9025 ?          S          0:01 xterm -g 100x45+0+200
  9176 ?          S          0:00 xterm -g 100x45+0+200
 29854 ?          S          0:21 xterm -g 100x75+20+0 -fn \
    -B&H-LucidaTypewriter-Medium-R-Normal-Sans-12-120-75-75-M-70-iso10646-1
  4378 ?          S          0:01 xterm -bg MistyRose1 -T root -n root -e su -l
 25543 ?          S          0:02 xterm -g 100x45+0+200
 22161 ?          R          0:14 xterm -g 100x45+0+200
 16832 ?          S          0:01 xterm -bg MistyRose1 -T root -n root -e su -l
 16912 ?          S          0:00 xterm -g 100x45+0+200
 17861 ?          S          0:00 xterm -bg DarkSeaGreen1 -g 120x45+40+300
 19930 ?          S          0:13 xterm -bg LightCyan
 21686 ?          S          0:04 xterm -g 100x45+0+200 -fn \
lucidasanstypewriter-12
 23104 ?          S          0:00 xterm -g 100x45+0+200
 26547 ?          S          0:00 xterm -g 100x45+0+200
```

程序清單可依照您的需要進行格式化。選項 `-L` 會傳回所有關鍵字의清單。輸入下列指令來發出一份按記憶體的使用排序的程序清單：

```
$ ps ax --format pid,rss,cmd --sort rss
  PID  RSS  CMD
    2     0 [ksoftirqd/0]
    3     0 [events/0]
   17     0 [kblockd/0]
[... ]
10164 5260 xterm
31110 5300 xterm
17010 5356 xterm
 3896 29292 /usr/X11R6/bin/X -nolisten tcp -br vt7 -auth /var/lib/xdm/authdir/au
```

36.8 程序樹：pstree

指令 `pstree` 會產生一份樹狀程序清單：

```

$ pstree
init--atd
  |-3*[automount]
  |-bdflush
  |-cron
  [...]
  |-usb-storage-1
  |-usb-storage-2
  |-10*[xterm---zsh]
  |-xterm---zsh---mutt
  |-2*[xterm---su---zsh]
  |-xterm---zsh---ssh
  |-xterm---zsh---pstree
  |-ypbind---ypbind---2*[ypbind]
  \-zsh---startx---xinit4--X
                                \-ctwm--xclock
                                    |-xload
                                        \-xosview.bin

```

參數 `-p` 會將程序ID 新增至指定的名稱。如果要同時顯示指令行，請使用 `-a` 參數：

```

$ pstree -pa
init,1
  |-atd,1255
  [...]
  \-zsh,1404
      \-startx,1407 /usr/X11R6/bin/startx
          \-xinit4,1419 /suse/jj/.xinitrc [...]
              |-X,1426 :0 -auth /suse/jj/.Xauthority
                  \-ctwm,1440
                      |-xclock,1449 -d -geometry -0+0 -bg grey
                          |-xload,1450 -scale 2
                              \-xosview.bin,1451 +net -bat +net

```

36.9 功能介紹：w

使用指令 `w`，可清楚掌握登入系統的使用者和他們的動作。例如：

```

$ w
15:17:26 up 62 days, 4:33, 14 users, load average: 0.00, 0.04, 0.01
USER      TTY      LOGIN@   IDLE   JCPU   PCPU   WHAT
jj        pts/0    30Mar04  4days 0.50s  0.54s xterm -e su -l
jj        pts/1    23Mar04  5days 0.20s  0.20s -zsh
jj        pts/2    23Mar04  5days 1.28s  1.28s -zsh

```

```

jj      pts/3      23Mar04  3:28m  3.21s  0.50s  -zsh
[... ]
jj      pts/7      07Apr04  0.00s  9.02s  0.01s  w
jj      pts/9      25Mar04  3:24m  7.70s  7.38s  mutt
[... ]
jj      pts/14     12:49    37:34   0.20s  0.13s  ssh totan

```

最末一行顯示使用者jj已建立一個安全的外圍程序(ssh)連線至電腦totan。如果其它系統的使用者從遠端登入，參數-f會顯示該使用者用來連線的電腦。

36.10 記憶體的使用：free

公用程式free會檢查RAM的使用狀況。顯示可使用和已使用的記憶體(及交換區域)的詳細資訊：

```

$ free

```

	total	used	free	shared	buffers	cached
Mem:	514736	273964	240772	0	35920	42328
-/+ buffers/cache:		195716	319020			
Swap:	1794736	104096	1690640			

使用-m，會以MB為單位來顯示大小：

```

$ free -m

```

	total	used	free	shared	buffers	cached
Mem:	502	267	235	0	35	41
-/+ buffers/cache:		191	311			
Swap:	1752	101	1651			

下面這一行包含相當有趣的資訊：

```

-/+ buffers/cache:      191      311

```

它可以計算緩衝區和快取所佔用的記憶體空間。參數-d延遲會確定每隔(延遲)秒自動更新顯示內容。例如，free -d 1.5每1.5秒會更新一次。

36.11 核心環狀緩衝區：dmesg

Linux 核心會將某些訊息保存在環狀緩衝區內。如果要檢視這些訊息，可輸入指令dmesg：

```
$ dmesg
[...]
sdc : READ CAPACITY failed.
sdc : status = 1, message = 00, host = 0, driver = 08
Info fld=0xa00 (nonstd), Current sd00:00: sense key Not Ready
sdc : block size assumed to be 512 bytes, disk size 1GB.
sdc: test WP failed, assume Write Enabled
sdc: I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
unable to read partition table
I/O error: dev 08:20, sector 0
nfs: server totan not responding, still trying
nfs: server totan OK
```

最後一行表示NFS 伺服器totan 中有一個暫時性的問題。在此之前的幾行是由插入USB 快閃磁碟機所觸發的。較早的事件記錄在檔案/var/log/messages 和/var/log/warn 中。

36.12 檔案系統及其使用：mount、df 和du

指令mount 會顯示哪個檔案系統(設備和類型) 已裝載於哪個裝載點：

```
$ mount
/dev/hdb2 on / type ext2 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hdal on /data type ext2 (rw)
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
automount(pid1012) on /suse type autofs \
(rw,fd=5,pgrp=1012,minproto=2,maxproto=3)
totan:/real-home/jj on /suse/jj type nfs \
(rw,nosuid,rsize=8192,wsiz=8192,hard,intr,nolock,addr=10.10.0.1)
```

使用指令 `df` 來取得檔案系統總使用量的資訊。參數 `-h` (或 `--human-readable`) 可將輸出轉換為一般使用者容易了解的形式。

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdb2       7.4G  5.1G  2.0G  73% /
/dev/hda1       74G   5.8G  65G   9% /data
shmfs           252M   0    252M  0% /dev/shm
totan:/real-home/jj 350G  324G  27G  93% /suse/jj
```

NFS 檔案伺服器 `totan` 的使用者應立刻清除他們的主目錄。可使用指令 `du` 來顯示指定目錄及其子目錄下所有檔案的大小。參數 `-s` 可以不顯示輸出的詳細資訊。`-h` 同樣可以將資料轉換為一般人容易了解的形式。使用這個指令：

```
$ du -sh ~
361M    /suse/jj
```

檢視您的主目錄佔用的空間。

36.13 /proc 檔案系統

`/proc` 檔案系統是一個虛擬檔案系統，核心會在此系統內以虛擬檔案的形式來保存重要資訊。例如，使用這個指令可顯示 CPU 類型。

```
$ cat /proc/cpuinfo
processor       : 0
vendor_id     : AuthenticAMD
cpu family    : 6
model         : 8
model name    : AMD Athlon(tm) XP 2400+
stepping      : 1
cpu MHz       : 2009.343
cache size    : 256 KB
fdiv_bug      : no
[...]
```

可用下列指令來查詢配置及中斷的使用：

```
$ cat /proc/interrupts
          CPU0
0: 537544462          XT-PIC  timer
1:   820082          XT-PIC  keyboard
```

```

2:          0          XT-PIC  cascade
8:          2          XT-PIC  rtc
9:          0          XT-PIC  acpi
10:         13970       XT-PIC  usb-uhci, usb-uhci
11:    146467509       XT-PIC  ehci_hcd, usb-uhci, eth0
12:     8061393       XT-PIC  PS/2 Mouse
14:     2465743       XT-PIC  ide0
15:     1355         XT-PIC  ide1
NMI:        0
LOC:        0
ERR:        0
MIS:        0

```

以下為一些重要檔案及其內容：

/proc/devices 可用設備

/proc/modules 載入的核心模組

/proc/cmdline 核心指令行

/proc/meminfo 有關記憶體使用狀況的詳細資訊

/proc/config.gz gzip-壓縮核心目前正在執行的組態檔案

如需更多資訊，請參閱 `/usr/src/linux/Documentation/filesystems/proc.txt`。有關執行中處理程序的資訊，請參閱 `/proc/<NNN>` 目錄，其中 `<NNN>` 為相關程序的程序ID (PID)。在 `/proc/self/` 中可以找到每個程序的特性：

```
$ ls -l /proc/self
lrwxrwxrwx 1 root root 64 Apr 29 13:52 /proc/self -> 27585
```

```
$ ls -l /proc/self/
total 0
dr-xr-xr-x  2 jj suse 0 Apr 29 13:52 attr
-r-----  1 jj suse 0 Apr 29 13:52 auxv
-r--r--r--  1 jj suse 0 Apr 29 13:52 cmdline
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 cwd -> /suse/jj/t
-r--r--r--  1 jj suse 0 Apr 29 13:52 delay
-r-----  1 jj suse 0 Apr 29 13:52 environ
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 exe -> /bin/ls
dr-x-----  2 jj suse 0 Apr 29 13:52 fd
-rw-----  1 jj suse 0 Apr 29 13:52 mapped_base
-r--r--r--  1 jj suse 0 Apr 29 13:52 maps
-rw-----  1 jj suse 0 Apr 29 13:52 mem
-r--r--r--  1 jj suse 0 Apr 29 13:52 mounts
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 root -> /
-r--r--r--  1 jj suse 0 Apr 29 13:52 stat
-r--r--r--  1 jj suse 0 Apr 29 13:52 statm
-r--r--r--  1 jj suse 0 Apr 29 13:52 status
dr-xr-xr-x  3 jj suse 0 Apr 29 13:52 task
-r--r--r--  1 jj suse 0 Apr 29 13:52 wchan

```

maps 檔中含有可執行檔和程式庫的位址指定：

```
$ cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:02 22890      /bin/cat
0804c000-0804d000 rw-p 00003000 03:02 22890      /bin/cat
0804d000-0806e000 rwxp 0804d000 00:00 0
40000000-40016000 r-xp 00000000 03:02 10882      /lib/ld-2.3.3.so
40016000-40017000 rw-p 00015000 03:02 10882      /lib/ld-2.3.3.so
40017000-40018000 rw-p 40017000 00:00 0
4002b000-40135000 r-xp 00000000 03:02 10908      /lib/tls/libc.so.6
40135000-4013d000 rw-p 0010a000 03:02 10908      /lib/tls/libc.so.6
4013d000-40141000 rw-p 4013d000 00:00 0
bffffe000-c0000000 rw-p bffffe000 00:00 0
fffffe000-fffff000 ---p 00000000 00:00 0
```

36.14 vmstat、iostat 及mpstat

公用程式vmstat會報告虛擬記憶體的統計數據。它會讀取檔案/proc/meminfo、/proc/stat以及/proc/*/stat。適合用來找出系統效能瓶頸。

指令iostat可以報告CPU的統計數據以及設備與分割區的輸出與輸入。顯示的資訊來自於/proc/stat檔案與/proc/partitions檔案。輸出可用來改善硬碟間輸出與輸入的負載平衡。指令mpstat會報告CPU的相關統計數據。

36.15 procinfo

使用指令procinfo來摘述/proc檔案系統的重要資訊：

```
$ procinfo
Linux 2.6.4-54.5-default (geeko@buildhost) (gcc 3.3.3 ) #1 1CPU [roth.suse.de]
```

Memory:	Total	Used	Free	Shared	Buffers
Mem:	516696	513200	3496	0	43284
Swap:	530136	1352	528784		

```
Bootup: Wed Jul 7 14:29:08 2004      Load average: 0.07 0.04 0.01 1/126 5302
```

user :	2:42:28.08	1.3%	page in :	0
nice :	0:31:57.13	0.2%	page out:	0
system:	0:38:32.23	0.3%	swap in :	0
idle :	3d 19:26:05.93	97.7%	swap out:	0
uptime:	4d 0:22:25.84		context :	207939498

irq 0:	776561217 timer	irq 8:	2 rtc
irq 1:	276048 i8042	irq 9:	24300 VIA8233
irq 2:	0 cascade [4]	irq 11:	38610118 acpi, eth0, uhci_hcd
irq 3:	3	irq 12:	3435071 i8042
irq 4:	3	irq 14:	2236471 ide0
irq 6:	2	irq 15:	251 ide1

如果要檢視所有資訊，請使用參數-a。參數-nN 每隔(N) 秒即更新資訊。在此例中，請按Ⓚ來終止程式。

依照預設，會顯示累計的值。參數-d 會產生差值。procinfo -dn5 會顯示過去5 秒內變更的值。

```
Memory:      Total      Used      Free      Shared      Buffers      Cached
Mem:         0          2         -2         0           0            0
Swap:        0          0          0

Bootup: Wed Feb 25 09:44:17 2004      Load average: 0.00 0.00 0.00 1/106 31902

user  :      0:00:00.02   0.4%  page in :      0  disk 1:      0r      0w
nice  :      0:00:00.00   0.0%  page out:     0  disk 2:      0r      0w
system: 0:00:00.00   0.0%  swap in :      0  disk 3:      0r      0w
idle  :      0:00:04.99 99.6%  swap out:     0  disk 4:      0r      0w
uptime: 64d 3:59:12.62      context :    1087

irq 0:      501 timer                irq 10:      0  usb-uhci, usb-uhci
irq 1:      1  keyboard          irq 11:      32 ehci_hcd, usb-uhci,
irq 2:      0  cascade [4]       irq 12:      132 PS/2 Mouse
irq 6:      0
irq 8:      0  rtc                irq 14:      0  ide0
irq 9:      0  acpi              irq 15:      0  ide1
```

36.16 PCI 資源 : lspci

指令lspci 會列出PCI 資源：

```
$ lspci
00:00.0 Host bridge: VIA Technologies, Inc. \
  VT8366/A/7 [Apollo KT266/A/333]
00:01.0 PCI bridge: VIA Technologies, Inc. \
  VT8366/A/7 [Apollo KT266/A/333 AGP]
00:0b.0 Ethernet controller: Digital Equipment Corporation \
  DECchip 21140 [FasterNet] (rev 22)
00:10.0 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.1 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.2 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.3 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 82)
00:11.0 ISA bridge: VIA Technologies, Inc. VT8235 ISA Bridge
00:11.1 IDE interface: VIA Technologies, Inc. VT82C586/B/686A/B \
  PIPC Bus Master IDE (rev 06)
00:11.5 Multimedia audio controller: VIA Technologies, Inc. \
  VT8233 AC97 Audio Controller (rev 50)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. \
  MGA G550 AGP (rev 01)
```

使用-v 可產生更詳細的清單：


```
$ lspci -v
[...]
01:00.0 \
VGA compatible controller: Matrox Graphics, Inc. MGA G550 AGP (rev 01) \
(prog-if 00 [VGA])
Subsystem: Matrox Graphics, Inc. Millennium G550 Dual Head DDR 32Mb
Flags: bus master, medium devsel, latency 32, IRQ 10
Memory at d8000000 (32-bit, prefetchable) [size=32M]
Memory at da000000 (32-bit, non-prefetchable) [size=16K]
Memory at db000000 (32-bit, non-prefetchable) [size=8M]
Expansion ROM at <unassigned> [disabled] [size=128K]
Capabilities: <available only to root>
```

有關設備名稱解析的資訊，請參閱檔案 `/usr/share/pci.ids`。本檔中未列出的 PCI ID 標記為「不明設備」。

參數 `-vv` 可產生所有可供程式查詢的資訊。如果要檢視純數值，應使用參數 `-n`。

36.17 執行程式的系統呼叫：strace

公用程式 `strace` 讓您可以追蹤執行中處理程序的所有系統呼叫。將 `strace` 新增至指令行的開頭，以正常方式來輸入指令：

```
$ strace -e open ls

execve("/bin/ls", ["ls"], [/* 88 vars */]) = 0
uname({sys="Linux", node="edison", ...}) = 0
brk(0) = 0x805b000
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
= 0x40017000
open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=76333, ...}) = 0
old_mmap(NULL, 76333, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40018000
[...]
ioctl(1, SNDCTL_TMR_TIMEBASE or TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=53, ws_col=110, ws_xpixel=897, ws_ypixel=693}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
fstat64(3, {st_mode=S_IFDIR|0755, st_size=144, ...}) = 0
fcntl164(3, F_SETFD, FD_CLOEXEC) = 0
getdents64(3, /* 5 entries */, 4096) = 160
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 48), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
= 0x40018000
write(1, "ltrace-ls.txt myfile.txt strac...", 41) = 41
munmap(0x40018000, 4096) = 0
exit_group(0) = ?
```

例如，要追蹤所有試圖開啓某一特定檔案的動作，可使用下列指令：

```
$ strace -e open ls myfile.txt

open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
open("/lib/tls/librt.so.1", O_RDONLY) = 3
open("/lib/libacl.so.1", O_RDONLY) = 3
open("/lib/libselinux.so.1", O_RDONLY) = 3
open("/lib/tls/libc.so.6", O_RDONLY) = 3
open("/lib/tls/libpthread.so.0", O_RDONLY) = 3
open("/lib/libattr.so.1", O_RDONLY) = 3
open("/proc/mounts", O_RDONLY) = 3
[...]
open("/proc/filesystems", O_RDONLY) = 3
open("/proc/self/attr/current", O_RDONLY) = 4
```

如果要追蹤所有子程序，可使用參數-f。可充分掌控strace的行為和輸出格式。如需更多資訊，請參閱man strace。

36.18 一個程式執行階段的程式庫呼叫：ltrace

指令ltrace讓您可以追蹤處理程序的程式庫呼叫。這個指令的用法與strace相似。參數-c會輸出程式庫呼叫的數目和持續時間。

```
$ ltrace -c find /usr/share/doc
% time      seconds  usecs/call   calls      errors  syscall
-----
 86.27      1.071814    30          35327      write
10.15      0.126092    38          3297       getdents64
 2.33      0.028931    3          10208      lstat64
 0.55      0.006861    2           3122       1 chdir
 0.39      0.004890    3           1567       2 open
[...]
 0.00      0.000003    3           1          uname
 0.00      0.000001    1           1          time
-----
100.00      1.242403                    58269      3 total
```

36.19 指定需要的程式庫：ldd

指令ldd可用來釐清哪些程式庫將載入引數指定的動態可執行檔：

```
$ ldd /bin/ls
linux-gate.so.1 => (0xffffe000)
librt.so.1 => /lib/tls/librt.so.1 (0x4002b000)
libacl.so.1 => /lib/libacl.so.1 (0x40033000)
libselinux.so.1 => /lib/libselinux.so.1 (0x40039000)
libc.so.6 => /lib/tls/libc.so.6 (0x40048000)
libpthread.so.0 => /lib/tls/libpthread.so.0 (0x4015d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
libattr.so.1 => /lib/libattr.so.1 (0x4016d000)
```

靜態二進位不需要任何動態程式庫：

```
$ ldd /bin/sash
        not a dynamic executable
$ file /bin/sash
/bin/sash: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
for GNU/Linux 2.2.5, statically linked, stripped
```

36.20 其它有關ELF 二進位的資訊

可使用readelf 公用程式來讀取二進位的內容。甚至可將它用在為其它硬體架構而建立的ELF 檔案：

```
$ readelf --file-header /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                           ELF32
  Data:                               2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                             UNIX - System V
  ABI Version:                        0
  Type:                               EXEC (Executable file)
  Machine:                             Intel 80386
  Version:                             0x1
  Entry point address:                0x8049b40
  Start of program headers:           52 (bytes into file)
  Start of section headers:          76192 (bytes into file)
  Flags:                               0x0
  Size of this header:                 52 (bytes)
  Size of program headers:            32 (bytes)
  Number of program headers:          9
  Size of section headers:            40 (bytes)
  Number of section headers:          29
  Section header string table index:  26
```

36.21 程序間的通訊：ipcs

指令 `ipcs` 可產生一份使用中的IPC 資源清單：

```
$ ipcs
----- Shared Memory Segments -----
key      shmid      owner      perms      bytes      nattch     status
0x000027d9 5734403    toms       660        64528      2
0x00000000 5767172    toms       666        37044      2
0x00000000 5799941    toms       666        37044      2

----- Semaphore Arrays -----
key      semid      owner      perms      nsems
0x000027d9 0          toms       660        1

----- Message Queues -----
key      msqid      owner      perms      used-bytes  messages
```

36.22 使用time 進行時間管理

可利用 `time` 公用程式來決定指令所花費的時間。這個公用程式有兩種版本：做為外圍程序內建使用和做為程式(`/usr/bin/time`)使用。

```
$ time find . > /dev/null

real    0m4.051s
user    0m0.042s
sys     0m0.205s
```

Part V

附 錄

資訊來源和文件

適用於SUSE LINUX 系統的資訊來源範圍很廣。部份來源是SUSE 特有的，但是很多則是通用來源。部份已經可以在您的系統或安裝媒體取得，其他可以從網際網路存取。

SUSE 文件

您可以在RPM 套件`suselinux-userguide_en` 與 `suselinux-adminguide_en` 的HTML 或PDF 格式的手冊中，找到詳細資訊。在標準安裝中，手冊會位於 `/usr/share/doc/manual/` 目錄。您可以從「SUSE 說明中心」存取此資訊。

The Linux Documentation Project (LDP)

The Linux Documentation Project (請參閱<http://www.tldp.org/>) 是一個志願團體，他們製造關於Linux 的文件。The LDP 包含HOWTO、「常見問題集」以及指南，全部都已免費公開發行。

HOWTO 是逐步指示而且適用於一般使用者、系統管理員以及程式設計人員。例如，HOWTO 會敘述DHCP 伺服器的建立以及該注意的重點，而不會說明Linux 本身的安裝方式。通常，此類型的文件是說明一般性問題，所以適用於每一種版本。`howto` 套件包含ASCII 格式的HOWTO。偏好HTML 格式的使用者應該安裝`howtoenh`。

「常見問題」是郵寄清單中經常提出的特定問題及其解答的相關集合，例如，“什麼是LDAP？”或“什麼是RAID？”此類別中的文字一般相當短。

指南是一些文件，可以處理的主題比HOWTO 和「常見問題」更詳細。範例包括核心程式設計和網路管理。相關意見構思提供讀者更詳細的資訊。

部份的LDP文件也提供其他格式，例如PDF、單一和多重HTML網頁、PostScript以及SGML或XML原始檔。在某些案例中，也提供不同語言的翻譯。

Man 頁面和資訊頁面

man 頁面(手冊頁面)是指令、系統呼叫、檔案格式或其類似項目的說明文字。man 頁面一般分成多個段落，例如名稱、語法、說明、選項和檔案。

要顯示man 頁面，輸入man 後面加上指令的名稱，例如man ls，它會顯示ls 指令的說明內容。使用游標鍵，移動可見區域。Ⓞ 結束man。若要列印man 頁面(例如針對ls 指令)，請輸入像man -Tps | lpr 這樣的指令。如需關於man 指令的詳細資訊，請使用--help 選項或是man 的man 頁面(man man)。

部份文件也提供資訊格式，例如grep。請使用info grep 存取它。

資訊頁面遠比man 頁面詳細。它們分成不同的節點—頁面，可以使用資訊讀取器閱讀，與網頁瀏覽器非常相似。使用Ⓟ(上一頁)和Ⓝ(下一頁)，在資訊頁中瀏覽。Ⓞ 結束info。其他按鍵列示在info 文件(info info)。

man 頁面和資訊頁面都可以使用Konqueror 閱讀。在URL 行輸入man:<指令> 或info:<指令>，開啓想要的文件。

標準和規格

目前有各種來源，提供關於標準或規格的資訊。

www.linuxbase.org Free Standards Group 是非營利的獨立組織，促進發送免費軟體和開放原始碼軟體。該組織定義發送獨立標準，致力於達到此目標。多個標準的維護，例如重要的LSB (Linux Standard Base) 是由該組織監督的。

<http://www.w3.org> World Wide Web Consortium (W3C) 是特別知名的標準組織之一。它是在1994年10月由Tim Berners-Lee 所創立，致力於網路技術的標準化。W3C 促進開放、免授權以及獨立於製造商規格的傳播，例如HTML、XHTML 和XML。這些網路標準分四階段在工作小組中開發，並以W3C 建議(REC) 的形式，推向公眾。

<http://www.oasis-open.org> OASIS (Organization for the Advancement of Structured Information Standards) 是國際性論壇，專門開發網路安全、電子商務、商業交易、邏輯以及不同市場間的交互運作標準。

<http://www.ietf.org> (IETF) 是由研究者、網路設計人員、供應商和使用者等組成的一個活躍的國際性組織。它著重於網際網路架構的開發，以及透過通訊協定平穩地操作網際網路。

每一個IETF標準會發行爲RFC (Request for Comments) 文件，而且是免費提供的。目前有6種類型的RFC：建議的標準、草稿標準、網際網路標準、實驗通訊協定、資訊文件以及歷史標準。只有前3個(建議、草稿和完整)較精細的IETF標準(請參閱<http://www.ietf.org/rfc/rfc1796.txt>)。

<http://www.ieee.org> Institute of Electrical and Electronics Engineers (IEEE) 是草擬資訊技術、電訊、醫藥與保健、運輸和其他等等區域標準的組織。IEEE標準需要付費。

<http://www.iso.org> ISO Committee (International Organization for Standards) 是全球最大的標準開發商，它的網路含蓋140個國家的標準機構。ISO標準需要付費。

<http://www.din.de>, <http://www.din.com>

Deutsches Institut für Normung (DIN) 是註冊的技術和科學協會。它成立於1917。根據DIN，該組織是“負責德國在全球和歐洲標準組織的利益。”

該協會讓製造商、客戶、貿易專家、服務公司、科學家和其他對標準建立感興趣的人聚集一起。該標準需要付費而且可以使用DIN首頁訂購。

檔案系統檢查

Manual Page of reiserfsck

REISERFSCK(8)

REISERFSCK(8)

NAME

reiserfsck - check a Linux Reiserfs file system

SYNOPSIS

```
reiserfsck [ -afprVy ] [ --rebuild-sb | --check | --fix-  
fixable | --rebuild-tree | --clean-attributes ] [ -j |  
--journal device ] [ -z | --adjust-size ] [ -n | --nolog ]  
[ -l | --logfile file ] [ -q | --quiet ] [ -y | --yes ] [  
-S | --scan-whole-partition ] [ --no-journal-available ]  
device
```

DESCRIPTION

Reiserfsck searches for a Reiserfs filesystem on a device, replays any necessary transactions, and either checks or repairs the file system.

device is the special file corresponding to the device or partition (e.g /dev/hdXX for IDE disk partition or /dev/sdXX for SCSI disk partition).

OPTIONS

--rebuild-sb

This option recovers the superblock on a Reiserfs partition. Normally you only need this option if mount reports "read_super_block: can't find a reiserfs file system" and you are sure that a Reiserfs file system is there.

--check

This default action checks file system consistency

and reports but does not repair any corruption that it finds. This option may be used on a read-only file system mount.

--fix-fixable

This option recovers certain kinds of corruption that do not require rebuilding the entire file system tree (`--rebuild-tree`). Normally you only need this option if the `--check` option reports "corruption that can be fixed with `--fix-fixable`". This includes: zeroing invalid data-block pointers, correcting `st_size` and `st_blocks` for directories, and deleting invalid directory entries.

--rebuild-tree

This option rebuilds the entire file system tree using leaf nodes found on the device. Normally you only need this option if the `--check` option reports "corruption that can be fixed only during `--rebuild-tree`". You are strongly encouraged to make a backup copy of the whole partition before attempting the `--rebuild-tree` option.

--clean-attributes

This option cleans reserved fields of Stat-Data items.

--journal device , -j device

This option supplies the device name of the current file system journal. This option is required when the journal resides on a separate device from the main data device (although it can be avoided with the expert option `--no-journal-available`).

--adjust-size, -z

This option causes `reiserfsck` to correct file sizes that are larger than the offset of the last discovered byte. This implies that holes at the end of a file will be removed. File sizes that are smaller than the offset of the last discovered byte are corrected by `--fix-fixable`.

--logfile file, -l file

This option causes `reiserfsck` to report any corruption it finds to the specified log file rather than `stderr`.

--nolog, -n

This option prevents `reiserfsck` from reporting any kinds of corruption.

- `--quiet, -q`
This option prevents reiserfsck from reporting its rate of progress.
- `--yes, -y`
This option inhibits reiserfsck from asking you for confirmation after telling you what it is going to do, assuming yes. For safety, it does not work with the `--rebuild-tree` option.
- `-a, -p` These options are usually passed by `fsck -A` during the automatic checking of those partitions listed in `/etc/fstab`. These options cause reiserfsck to print some information about the specified file system, check if error flags in the superblock are set and do some light-weight checks. If these checks reveal a corruption or the flag indicating a (possibly fixable) corruption is found set in the superblock, then reiserfsck switches to the fixable mode. If the flag indicating a fatal corruption is found set in the superblock, then reiserfsck finishes with an error.
- `-V` This option prints the reiserfsprogs version and exit.
- `-r, -f` These options are ignored.

EXPERT OPTIONS

DO NOT USE THESE OPTIONS UNLESS YOU KNOW WHAT YOU ARE DOING. WE ARE NOT RESPONSIBLE IF YOU LOSE DATA AS A RESULT OF THESE OPTIONS.

- `--no-journal-available`
This option allows reiserfsck to proceed when the journal device is not available. This option has no effect when the journal is located on the main data device. NOTE: after this operation you must use reiserfstune to specify a new journal device.
- `--scan-whole-partition, -S`
This option causes `--rebuild-tree` to scan the whole partition, not only used space on the partition.

EXAMPLE OF USING

1. You think something may be wrong with a reiserfs partition on `/dev/hda1` or you would just like to perform a periodic disk check.

2. Run `reiserfsck --check --logfile check.log /dev/hda1`. If `reiserfsck --check` exits with status 0 it means no errors were discovered.

3. If `reiserfsck --check` exits with status 1 (and reports about fixable corruptions) it means that you should run `reiserfsck --fix-fixable --logfile fixable.log /dev/hda1`.

4. If `reiserfsck --check` exits with status 2 (and reports about fatal corruptions) it means that you need to run `reiserfsck --rebuild-tree`. If `reiserfsck --check` fails in some way you should also run `reiserfsck --rebuild-tree`, but we also encourage you to submit this as a bug report.

5. Before running `reiserfsck --rebuild-tree`, please make a backup of the whole partition before proceeding. Then run `reiserfsck --rebuild-tree --logfile rebuild.log /dev/hda1`.

6. If the `--rebuild-tree` step fails or does not recover what you expected, please submit this as a bug report. Try to provide as much information as possible and we will try to help solve the problem.

EXIT CODES

`reiserfsck` uses the following exit codes:

- 0 - No errors.
- 1 - File system errors corrected.
- 4 - File system fatal errors left uncorrected,
`reiserfsck --rebuild-tree` needs to be launched.
- 6 - File system fixable errors left uncorrected,
`reiserfsck --fix-fixable` needs to be launched.
- 8 - Operational error.
- 16 - Usage or syntax error.

AUTHOR

This version of `reiserfsck` has been written by Vitaly Fertman <vitaly@namesys.com>.

BUGS

There are likely to be some bugs. Please report bugs to the ReiserFS mail-list <reiserfs-list@namesys.com>.

TODO

Faster recovering, signal handling, i/o error handling, etc.

SEE ALSO

`mkreiserfs(8)`, `reiserfstune(8)` `resize_reiserfs(8)`, `debu`

greiserfs(8),

Reiserfsprogs-3.6.9

April 2003

REISERFSCK(8)

Manual Page of e2fsck

E2FSCK(8)

E2FSCK(8)

NAME

e2fsck - check a Linux second extended file system

SYNOPSIS

```
e2fsck [ -pacnyrdfvstDFSV ] [ -b superblock ] [ -B block
size ] [ -l|-L bad_blocks_file ] [ -C fd ] [ -j external-
journal ] [ -E extended_options ] device
```

DESCRIPTION

e2fsck is used to check a Linux second extended file system (ext2fs). E2fsck also supports ext2 filesystems containing a journal, which are also sometimes known as ext3 filesystems, by first applying the journal to the filesystem before continuing with normal e2fsck processing. After the journal has been applied, a filesystem will normally be marked as clean. Hence, for ext3 filesystems, e2fsck will normally run the journal and exit, unless its superblock indicates that further checking is required.

device is the device file where the filesystem is stored (e.g. /dev/hdc1).

OPTIONS

-a This option does the same thing as the -p option. It is provided for backwards compatibility only; it is suggested that people use -p option whenever possible.

-b superblock

Instead of using the normal superblock, use an alternative superblock specified by superblock. This option is normally used when the primary superblock has been corrupted. The location of the backup superblock is dependent on the filesystem's blocksize. For filesystems with 1k blocksizes, a backup superblock can be found at block 8193; for filesystems with 2k blocksizes, at block 16384; and

for 4k block sizes, at block 32768.

Additional backup superblocks can be determined by using the mke2fs program using the -n option to print out where the superblocks were created. The -b option to mke2fs, which specifies block size of the filesystem must be specified in order for the superblock locations that are printed out to be accurate.

If an alternative superblock is specified and the filesystem is not opened read-only, e2fsck will make sure that the primary superblock is updated appropriately upon completion of the filesystem check.

-B blocksize

Normally, e2fsck will search for the superblock at various different block sizes in an attempt to find the appropriate block size. This search can be fooled in some cases. This option forces e2fsck to only try locating the superblock at a particular block size. If the superblock is not found, e2fsck will terminate with a fatal error.

-c

This option causes e2fsck to run the badblocks(8) program to find any blocks which are bad on the filesystem, and then marks them as bad by adding them to the bad block inode. If this option is specified twice, then the bad block scan will be done using a non-destructive read-write test.

-C fd

This option causes e2fsck to write completion information to the specified file descriptor so that the progress of the filesystem check can be monitored. This option is typically used by programs which are running e2fsck. If the file descriptor specified is 0, e2fsck will print a completion bar as it goes about its business. This requires that e2fsck is running on a video console or terminal.

-d

Print debugging output (useless unless you are debugging e2fsck).

-D

Optimize directories in filesystem. This option causes e2fsck to try to optimize all directories, either by reindexing them if the filesystem supports directory indexing, or by sorting and compressing directories for smaller directories, or

- for filesystems using traditional linear directories.
- E `extended_options`
Set `e2fsck` extended options. Extended options are comma separated, and may take an argument using the equals (`'='`) sign. The following options are supported:
 - `ea_ver=extended_attribute_version`
Assume the format of the extended attribute blocks in the filesystem is the specified version number. The version number may be 1 or 2. The default extended attribute version format is 2.
 - f Force checking even if the file system seems clean.
 - F Flush the filesystem device's buffer caches before beginning. Only really useful for doing `e2fsck` time trials.
 - j `external-journal`
Set the pathname where the external-journal for this filesystem can be found.
 - l `filename`
Add the block numbers listed in the file specified by `filename` to the list of bad blocks. The format of this file is the same as the one generated by the `badblocks(8)` program. Note that the block numbers are based on the blocksize of the filesystem. Hence, `badblocks(8)` must be given the blocksize of the filesystem in order to obtain correct results. As a result, it is much simpler and safer to use the `-c` option to `e2fsck`, since it will assure that the correct parameters are passed to the `badblocks` program.
 - L `filename`
Set the bad blocks list to be the list of blocks specified by `filename`. (This option is the same as the `-l` option, except the bad blocks list is cleared before the blocks listed in the file are added to the bad blocks list.)
 - n Open the filesystem read-only, and assume an answer of `'no'` to all questions. Allows `e2fsck` to be used non-interactively. (Note: if the `-c`, `-l`, or `-L` options are specified in addition to the `-n` option,

then the filesystem will be opened read-write, to permit the bad-blocks list to be updated. However, no other changes will be made to the filesystem.)

- p Automatically repair ("preen") the file system without any questions.
- r This option does nothing at all; it is provided only for backwards compatibility.
- s This option will byte-swap the filesystem so that it is using the normalized, standard byte-order (which is i386 or little endian). If the filesystem is already in the standard byte-order, e2fsck will take no action.
- S This option will byte-swap the filesystem, regardless of its current byte-order.
- t Print timing statistics for e2fsck. If this option is used twice, additional timing statistics are printed on a pass by pass basis.
- v Verbose mode.
- V Print version information and exit.
- y Assume an answer of 'yes' to all questions; allows e2fsck to be used non-interactively.

EXIT CODE

The exit code returned by e2fsck is the sum of the following conditions:

- 0 - No errors
- 1 - File system errors corrected
- 2 - File system errors corrected, system should be rebooted
- 4 - File system errors left uncorrected
- 8 - Operational error
- 16 - Usage or syntax error
- 32 - E2fsck canceled by user request
- 128 - Shared library error

SIGNALS

The following signals have the following effect when sent to e2fsck.

SIGUSR1

This signal causes e2fsck to start displaying a completion bar. (See discussion of the -C option.)

SIGUSR2

This signal causes e2fsck to stop displaying a completion bar.

REPORTING BUGS

Almost any piece of software will have bugs. If you manage to find a filesystem which causes e2fsck to crash, or which e2fsck is unable to repair, please report it to the author.

Please include as much information as possible in your bug report. Ideally, include a complete transcript of the e2fsck run, so I can see exactly what error messages are displayed. If you have a writeable filesystem where the transcript can be stored, the script(1) program is a handy way to save the output of e2fsck to a file.

It is also useful to send the output of dumpe2fs(8). If a specific inode or inodes seems to be giving e2fsck trouble, try running the debugfs(8) command and send the output of the stat(1u) command run on the relevant inode(s). If the inode is a directory, the debugfs dump command will allow you to extract the contents of the directory inode, which can sent to me after being first run through uen code(1).

Always include the full version string which e2fsck displays when it is run, so I know which version you are running.

AUTHOR

This version of e2fsck was written by Theodore Ts'o <tytso@mit.edu>.

SEE ALSO

mke2fs(8), tune2fs(8), dumpe2fs(8), debugfs(8)

E2fsprogs version 1.34

July 2003

E2FCK(8)

Manual Page of xfs_check

xfs_check(8)

xfs_check(8)

NAME

xfs_check - check XFS filesystem consistency

SYNOPSIS

xfs_check [-i ino] ... [-b bno] ... [-s] [-v] xfs_special

xfs_check -f [-i ino] ... [-b bno] ... [-s] [-v] file

DESCRIPTION

xfs_check checks whether an XFS filesystem is consistent. It is normally run only when there is reason to believe that the filesystem has a consistency problem. The filesystem to be checked is specified by the xfs_special argument, which should be the disk or volume device for the filesystem. Filesystems stored in files can also be checked, using the -f flag. The filesystem should normally be unmounted or read-only during the execution of xfs_check. Otherwise, spurious problems are reported.

The options to xfs_check are:

- f Specifies that the special device is actually a file (see the mkfs.xfs -d file option). This might happen if an image copy of a filesystem has been made into an ordinary file.
- s Specifies that only serious errors should be reported. Serious errors are those that make it impossible to find major data structures in the filesystem. This option can be used to cut down the amount of output when there is a serious problem, when the output might make it difficult to see what the real problem is.
- v Specifies verbose output; it is impossibly long for a reasonably-sized filesystem. This option is intended for internal use only.
- i ino Specifies verbose behavior for a specific inode. For instance, it can be used to locate all the blocks associated with a given inode.
- b bno Specifies verbose behavior for a specific filesystem block. For instance, it can be used to determine what a specific block is used for. The block number is a "file system block number". Conversion between disk addresses (i.e. addresses reported by xfs_bmap) and file system blocks may be accomplished using xfs_db's convert command.

Any non-verbose output from `xfs_check` means that the filesystem has an inconsistency. The filesystem can be repaired using either `xfs_repair(8)` to fix the filesystem in place, or by using `xfsdump(8)` and `mkfs.xfs(8)` to dump the filesystem, make a new filesystem, then use `xfsrestore(8)` to restore the data onto the new filesystem. Note that `xfsdump` may fail on a corrupt filesystem. However, if the filesystem is mountable, `xfsdump` can be used to try and save important data before repairing the filesystem with `xfs_repair`. If the filesystem is not mountable though, `xfs_repair` is the only viable option.

DIAGNOSTICS

Under one circumstance, `xfs_check` unfortunately might dump core rather than produce useful output. If the filesystem is completely corrupt, a core dump might be produced instead of the message `xxx is not a valid filesystem`

If the filesystem is very large (has many files) then `xfs_check` might run out of memory. In this case the message out of memory is printed.

The following is a description of the most likely problems and the associated messages. Most of the diagnostics produced are only meaningful with an understanding of the structure of the filesystem.

`agf_freeblks n, counted m in ag a`

The freeblocks count in the allocation group header for allocation group a doesn't match the number of blocks counted free.

`agf_longest n, counted m in ag a`

The longest free extent in the allocation group header for allocation group a doesn't match the longest free extent found in the allocation group.

`agi_count n, counted m in ag a`

The allocated inode count in the allocation group header for allocation group a doesn't match the number of inodes counted in the allocation group.

`agi_freecount n, counted m in ag a`

The free inode count in the allocation group header for allocation group a doesn't match the number of inodes counted free in the allocation group.

`block a/b expected inum 0 got i`

The block number is specified as a pair (allocation

group number, block in the allocation group). The block is used multiple times (shared), between multiple inodes. This message usually follows a message of the next type.

block a/b expected type unknown got y
The block is used multiple times (shared).

block a/b type unknown not expected
The block is unaccounted for (not in the freelist and not in use).

link count mismatch for inode nnn (name xxx), nlink m, counted n
The inode has a bad link count (number of references in directories).

rtblock b expected inum 0 got i
The block is used multiple times (shared), between multiple inodes. This message usually follows a message of the next type.

rtblock b expected type unknown got y
The real-time block is used multiple times (shared).

rtblock b type unknown not expected
The real-time block is unaccounted for (not in the freelist and not in use).

sb_fdblocks n, counted m
The number of free data blocks recorded in the superblock doesn't match the number counted free in the filesystem.

sb_frextents n, counted m
The number of free real-time extents recorded in the superblock doesn't match the number counted free in the filesystem.

sb_icount n, counted m
The number of allocated inodes recorded in the superblock doesn't match the number allocated in the filesystem.

sb_ifree n, counted m
The number of free inodes recorded in the superblock doesn't match the number free in the filesystem.

SEE ALSO

mkfs.xfs(8), xfsdump(8), xfsrestore(8), xfs_ncheck(8),

xfs_repair(8), xfs(5).

xfs_check(8)

Manual Page of jfs_fsck

jfs_fsck(8) JFS utility - file system check jfs_fsck(8)

NAME

jfs_fsck - initiate replay of the JFS transaction log, and check and repair a JFS formatted device

SYNOPSIS

```
jfs_fsck [ -afnpvV ] [ -j journal_device ] [ --omit_jour-  
nal_replay ] [ --replay_journal_only ] device
```

DESCRIPTION

jfs_fsck is used to replay the JFS transaction log, check a JFS formatted device for errors, and fix any errors found.

device is the special file name corresponding to the actual device to be checked (e.g. /dev/hdb1).

jfs_fsck must be run as root.

WARNING

jfs_fsck should only be used to check an unmounted file system or a file system that is mounted READ ONLY. Using jfs_fsck to check a file system mounted other than READ ONLY could seriously damage the file system!

OPTIONS

If no options are selected, the default is -p.

- a Autocheck mode - Replay the transaction log. Do not continue fsck processing unless the aggregate state is dirty or the log replay failed. Functionally equivalent to -p. Autocheck mode is typically the default mode used when jfs_fsck is called at boot time.
- f Replay the transaction log and force checking even if the file system appears clean. Repair all problems

automatically.

- j journal_device
Specify the journal device.
- n Open the file system read only. Do not replay the transaction log. Report errors, but do not repair them.
- omit_journal_replay
Omit the replay of the transaction log. This option should not be used unless as a last resort (i.e. the log has been severely corrupted and replaying it causes further problems).
- p Automatically repair ("preen") the file system. Replay the transaction log. Do not continue fsck processing unless the aggregate state is dirty or the log replay failed. Functionally equivalent to -a.
- replay_journal_only
Only replay the transaction log. Do not continue with a full file system check if the replay fails or if the file system is still dirty even after a journal replay. In general, this option should only be used for debugging purposes as it could leave the file system in an unmountable state. This option cannot be used with -f, -n, or --omit_journal_replay.
- v Verbose messaging - print details and debug statements to stdout.
- V Print version information and exit (regardless of any other chosen options).

EXAMPLES

Check the 3rd partition on the 2nd hard disk, print extended information to stdout, replay the transaction log, force complete jfs_fsck checking, and give permission to repair all errors:

```
jfs_fsck -v -f /dev/hdb3
```

Check the 5th partition on the 1st hard disk, and report, but do not repair, any errors:

```
jfs_fsck -n /dev/hda5
```

EXIT CODE

The exit code returned by `jfs_fsck` represents one of the following conditions:

- 0 No errors
- 1 File system errors corrected and/or transaction log replayed successfully
- 2 File system errors corrected, system should be rebooted if file system was mounted
- 4 File system errors left uncorrected
- 8 Operational error
- 16 Usage or syntax error
- 128 Shared library error

REPORTING BUGS

If you find a bug in JFS or `jfs_fsck`, please report it via the bug tracking system ("Report Bugs" section) of the JFS project web site:

<http://oss.software.ibm.com/jfs>

Please send as much pertinent information as possible, including the complete output of running `jfs_fsck` with the `-v` option on the JFS device.

SEE ALSO

`fsck(8)`, `jfs_mkfs(8)`, `jfs_fscklog(8)`, `jfs_tune(8)`, `jfs_log-dump(8)`, `jfs_debugfs(8)`

AUTHORS

Barry Arndt (barndt@us.ibm.com)
William Braswell, Jr.

`jfs_fsck` is maintained by IBM.
See the JFS project web site for more details:
<http://oss.software.ibm.com/jfs>

October 29, 2002

`jfs_fsck(8)`



GNU 通用公共授權 (General Public License)

GNU 通用公共授權 (General Public License)

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA

每個人皆可複製和散佈本授權文件的完整重製物，但不得變更其內容。

前言

大多數軟體授權的設計是剝奪您分享與修改軟體的自由。相反地，*GNU 通用公共授權 (General Public License)* 的目的是保證您可以自由分享和變更自由軟體——以確保所有人都可以自由使用自由軟體。本*通用公共授權 (General Public License)* 適用於大部份自由軟體基金會的軟體以及任何其作者認可使用的其他程式。(部份其他自由軟體基金會軟體則受*GNU 程式庫通用公共授權 (General Public License)* 的規定)。您也可以讓您的程式適用本授權規定。

當我們提到“自由”軟體，我們指的是自由，不是價格。我們的「通用公共授權 (General Public License)」是要確定您可以自由散佈自由軟體的副本(如果您希望，也可以對本服務的收費)，確保您能收到原始碼或者在您需要時便能得到它，確保您能變更軟體或將它的一部分用於新的自由軟體；並且確保您知道您可以從事上述的事情。

爲了保護您的權利，我們必須訂定一些規範，禁止任何人否認您擁有這些權利，或者要求您放棄權利。如果您散佈軟體的副本或者您修改它，這些限制會轉化爲您的特定責任。

例如，如果您散佈這樣的程式副本，無論免費或收費，必須將您所享有的一切權利交付收受者。您必須確定他們也會收到或者可以取得原始程式碼。而且您必須讓他們知道以上條款，讓他們知道該有的權利。

我們有兩項措施，保護您的權利：(1) 以著作權保護軟體，以及(2) 提供您本授權，賦與您複製、散佈和/或修改軟體的法律許可。

同時，為了保護作者與我們（自由軟體基金會），我們希望確定每個人都明瞭，自由軟體是沒有擔保責任的。如果軟體被他人修改並加以傳遞，我們需要其收受者知道，他們所得到的並非原軟體，因此由他人所引起的任何問題對原作者的聲譽將不會有任何的影響。

最後，任何自由軟體經常遭受到軟體專利權的威脅。我們希望能避免自由軟體的再散佈者，以個人名義取得專利授權而使程式專有化的風險。要防止此狀況，我們已經明確表示任何專利權必須授權給每一個人自由使用的權利，不然就不符合本授權規定。

複製、散佈和修改的明確條款和條件如下：

GNU 通用公共授權(General Public License)

複製、散佈和修改的條款和條件

0. 凡著作權人在其程式或其他著作中聲明，該程式或著作得以通用公共授權(General Public License) 條款散佈，本授權對其均可適用。以下所稱的“程式”，係指任何一種適用通用公共授權(General Public License) 的程式或著作，基於本程式所生的著作，則指本「程式」或任何基於著作權法所產生的衍生著作：換言之，係指包含程式全部或部分的著作，不論是完整或經過修改的程式，以及（或）翻譯成其他語言的程式（以下“修改”一詞包括但不限於翻譯行為在內）。被授權人則稱為“您”。

複製、散佈和修改之外的其他活動，不適用本「授權」；這些活動超出其範圍。而本「程式」的輸出只有在其內容構成基於本「程式」所生的著作（而非只是因為執行本「程式」所造成）時，始受本授權拘束。至於程式的輸出內容是否構成本「程式」的衍生著作，則取決於「程式」的實際用途。

1. 您可以對所收受的「本程式」原始程式碼，無論以何種媒介，複製與散佈其完整的重製物，然而您必須符合以下要件：以顯著及適當的方式在每一份重製物上發佈適當的著作權標示及無擔保聲明；維持所有有關本授權以及無擔保聲明的原貌；並將本授權的副本連同「本程式」一併交付予其他任一位本程式的收受者。

您可以針對傳送副本的實體行為收取費用，而且您可以自行決定是否提供擔保來交換費用。

2. 若符合以下條件，您就可以對「程式」的副本或者它的任何一個部分進行修改，從基於「本程式」所生的著作，並且按照「條款1」複製和散佈此類修改或著作。

1. 您必須在所修改的檔案上附加顯著的標示，說明您修改過這些檔案，以及修改日期。
2. 您必須就您所散佈或發行的著作，無論是包含「本程式」全部或部分的著作，或者是自「本程式」或其任何部份所衍生的著作，整體授權所有第三人依本授權規定使用，且不得因此項授權行為而收取任何費用。
3. 若經過修改的程式在執行時通常以互動方式讀取命令時，您必須在最常被使用的方式下，於開始進入這種互動式使用時，列印或展示以下宣告：適當的著作權標示及無擔保聲明（或者聲明您提供擔保）、使用者可以依這些條件再散佈此程式，以及告知使用者如何瀏覽本授權的複本。（例外：若本程式本身係以互動的方式執行，然而通常卻不會列印該聲明時，則您基於「本程式」所生的著作便無需列印該聲明。）

整體而言，修改過的著作適用於這些要求。如果著作中可識別的一部份並非衍生自「本程式」，並且可以合理地認為是獨立的、個別的著作，則當您將其作為個別著作散佈時，本授權及其條款將不適用於該部分。然而當您將上述部分，作為基於「本程式」所生著作的一部份而散佈時，整個著作的散佈必須符合本授權條款的規定，而本授權對於其他被授權人所為的許可適用於著作整體。

因此，本條規定的意圖不在於主張或剝奪您對於完全由您所完成著作的權利；應該說，本條規定意在行使對基於程式所生的之衍生著作或集合著作散佈行為的控制權。

此外，非基於本「程式」所生的其他著作與本程式（或基於本程式所生的著作）在同一儲存或散佈的媒介上的單純聚集行為，並不會使該著作因此受本授權條款拘束。

3. 您可依前述第一、二條規定，複製與散佈本程式（或第二條所述基於「本程式」所產生的著作）的目的碼或可執行形式，但您必須符合以下要件：

1. 附上完整、相對應的機器可判讀原始程式碼，而這些原始程式碼必須依前述第一、二條規定，在經常用以做為軟體交換的媒介物上散佈；或
2. 附上至少三年有效的書面報價文件，提供任何第三人在支付不超過實際散佈原始程式碼所需成本的費用下，取得相同原始程式碼的完整機器可讀重製物，並依前述第一、二條規定在經常用以作為軟體交換的媒介物上散佈該重製物；或

3. 附上您所收受有關散佈相同原始程式碼的報價資訊。（本項選擇僅在非營利散佈、且僅在您依前述b項方式自該書面報價文件收受程式物件程式碼或執行檔時，才適用。）

著作的原始程式碼，是指對著作進行修改時適用的形式。對於一個可執行的著作而言，“完整的原始程式碼”是指著作中所包含所有模組的全部原始程式碼，加上相關介面的定義檔，還加上用以控制該著作編譯與安裝的描述。然而，特別的例外情況是，所散佈的原始程式碼並不需包含任何通常會隨著所執行作業系統的主要組成部分（編譯器、核心等等）而散佈的軟體（無論以原始程式碼或二進位格式），除非該部分本身即附加在執行檔。

若可執行檔或目標程式碼的散佈方式，是以指定的地點提供存取位置供人複製，則提供可自相同地點複製原始程式碼的使用機會，視同對於原始程式碼的散佈，然而第三人並不因此而負有將目標程式碼連同原始程式碼一併複製的義務。

4. 本「授權」所明示的方式外，您不得對本程式加以複製、修改、再授權或散佈。任何試圖以其他方式進行複製、修改、再授權或者散佈本「程式」的行為均為無效，並且將自動終止您基於本授權所得享有的權利。然而，依本授權規定自您手中收受重製物或權利之人，只要遵守本授權規定，他們所獲得的授權並不會因此終止。

5. 因為您並未在本授權上簽名，所以您無需接受本授權。然而，除此之外您別無其他修改或散佈本程式或其衍生著作的授權許可。若您不接受本「授權」，則這些行為在法律上都是被禁止的。因此，藉由對本程式（或任何基於本「程式」所衍生的著作）的修改或散佈行為，表示您接受本授權，以及接受所有關於複製、散佈或修改本「程式」或基於本「程式」所衍生著作的條款與條件。

6. 當您再散佈本「程式」（或任何基於本「程式」所衍生的著作）時，收受者即自動獲得原授權人所授予依本授權條款與條件複製、散佈或修改本「程式」的權利。您不得就本授權所賦予收受者行使的權利附加任何進一步的限制。您對於第三人是否履行本「授權」一事，無須負責。

7. 若法院判決、專利侵權主張或者其他任何理由（不限於專利爭議）的結果，使得加諸於您的條件（無論是由法院命令、協議或其他方式造成）與本授權規定有所衝突，他們不能將您排除在本「授權」的規範之外。若您無法同時符合依本「授權」所生義務及其他相關義務而進行散佈，那麼其結果便是您不得散佈該「程式」。例如，若專利授權不允許直接或間接透過您而取得重製物之人，以免付權利金的方式再散佈該「程式」時，您唯一能同時滿足該義務及本「授權」的方式就是徹底避免進行該「程式」的散佈。

若本條任一部份在特殊情況下被認定無效或無法執行時，本條其餘部分仍應適用，且本條全部於其他情況下仍應適用。

本條的目的並不在誘使您侵害專利或其他財產權的權利主張，或就此類主張的有效性加以爭執；本條的唯一目的，是在保障藉由公共授權慣例所執行自由軟體散



佈系統的完整性。許多人信賴該系統一貫使用的應用程式，而對經由此系統散佈的大量軟體有相當多的貢獻；作者/貢獻者有權決定他或她是否希望經由其他的系統散佈軟體，而被授權人則無該種選擇權。

本條的用意在於將本「授權」其他不確定的部分徹底解釋清楚。

8. 若因為專利或享有著作權保護的介面問題，而使得本「程式」的散佈和/或使用侷限於某些國家時，則將本程式置於本「授權」規範之下的原著作權人得增列明確的散佈地區限制條款，將該等國家排除在外，而使散佈的許可只限在未受排除的國家之內或之中。在該等情況下，該限制條款如同以書面方式訂定於本授權內容中，而成為本「授權」的條款。

9. 自由軟體基金會可以隨時公佈修訂和/或新的通用公共授權(*General Public License*) 版本。類似的新版本，其精神和目前的版本類似，但解決新問題或事情的細節則不同。

每一個版本會提供截然不同的版本編號。如果「程式」指定本「授權」套用的版本編號以及“任何新的版本”，您可以使用自由軟體基金會公佈的該版本以及任何新版本的以下條款和條件。如果「程式」未指定本「許可證」的版本編號，您可以選擇自由軟體基金會公佈的任何版本。

10. 如果您佈望將「程式」的某些部份合併至分送條件不同的其他自由軟體，請洽詢作者，索取許可權。至於受自由軟體基金會版權保護的軟體，請洽詢自由軟體基金會；我們有時候會對此做出例外。我們的決定會由兩個目標引導，便是保留我們自由軟體，所有衍生著作的自由狀態，以及促進軟體的普遍性分享和重複使用。

無擔保聲明

11. 因為程式是免費授權的，因此就可援用法律的允許程度，並不提供程式的擔保。非經書面聲明，著作權人和/或其他提供程式之人，無論明示或默示，均係依“現況”提供本程式而並無任何形式的擔保責任，其包括但不限於，就適售性以及特定目的的適用性為默示性擔保。有關本程式品質與效能的全部風險悉由您承擔。如本程式被證明有瑕疵，您應承擔所有服務、修復或改正的費用。

12. 非經法律要求或書面同意，任何著作權人或任何可能依前述方式修改和/或散佈本程式者，對於您因為使用或不能使用本程式所造成的一般性、特殊性、意外性或間接性損失，不負任何責任（包括但不限於，資料損失，資料執行不精確，或應由您或第三人承擔的損失，或本「程式」無法與其他程式運作等），即便前述的著作權人或其他人已被告知該等損失的可能性時，亦同。

條款和條件結束

如何將這些條款套用至您的新程式

如果您開發新程式，而且您想儘可能推向最多數公眾使用，最好的方法是讓它成為自由軟體，讓每個人都可以在這些條款下轉送和變更。

要做到這一點，將以下通知加到程式中。將它們加到每一個原始檔的開頭，是最安全的作法，可以最有效的表達不提供擔保；而且每一個檔案至少應該有一個“版權”行以及本份聲明全文位置的提示。

<使用一行文字提供程式名稱及其功能的簡述。>

Copyright (C) <年> <作者名稱>

本程式是自由軟體；您可以按照「自由軟體基金會」的「GNU 通用公共授權 (General Public License)」

散佈本程式是基於希望它會有所幫助，不過「不提供任何擔保」；也不包含「銷售性」或「適用特殊用途」

您應該收到一份「GNU 通用公共授權 (General Public License)」以及本程式；如果尚未收到，

另外透過電子郵件或一般信件，提供您的連絡資訊。

如果程式是互動性的，請在互動式模式開始時，輸出簡短提示如下：

Gnomovision 69 版，版權所有 (C) <年> <作者名稱>

Gnomovision 是「絕不提供擔保/保證」；如需詳細資訊，請輸入 'show w'。這是自由軟體，而且歡迎

假設的指令 `show w` 和 `show c` 應該會顯示「通用公共授權 (General Public License)」的適當部份。當然，您可以使用 `show w` 和 `show c` 以外的指令名稱；甚至以滑鼠鍵擊或選單方式進行——只要適合您的程式。

你也應該讓雇主(如果您是程式設計人員)或者學校(如果有的話)，在需要時簽署程式的“著作權放棄承諾書”。這是一個範例；請更改名稱：

Yoyodyne 公司，茲此放棄 James Hacker 所寫之 “Gnomovision” 程式 (該程式經由編譯器的) 所

Ty Coon 的簽名, 1st April 1989 Ty Coon, President of Vice

本通用公共授權 (General Public License) 不允許將您的程式合併至專利程式。如果您的程式是常式程式庫，會發現允許將專利應用程式結合程式庫，幫助更大。如果這是您想做的，請使用 GNU 程式庫通用公共授權 (General Public License) 取代本「授權」。

詞彙

存取許可權

檔案的存取許可權決定使用者或群組是否可讀取、寫入或執行檔案或目錄。通常系統管理員會設定存取許可權。

帳戶

帳戶是由使用者名稱或登入名稱與密碼所定義。帳戶是與使用者ID (UID) 對應。

ACL (存取控制清單, Access Control List)

檔案與目錄基本許可權的延伸概念。這些可讓存取許可權獲得更精確地控制。

ADSL (非對稱數位用戶迴路, Asymmetric Digital Subscriber Line)

使用電話網路的快速傳輸協定。

AGP (繪圖加速埠, Accelerated Graphics Port)

供繪圖卡使用的高速插槽，提供比PCI 更快的頻寬。AGP 繪圖卡可以直接還原(不需路由至處理器) 到隨機存取記憶體。

ATAPI (AT 封包連接介面, Advanced Technology Attachment Packet Interface)

ATAPI 是一種連線到(E)IDE 控制器的光碟機介面類型。除了ATAPI 介面之外，還有由SCSI 控制器處理的SCSI 光碟機。

備份

備份是用來還原已經損毀或遺失的資料複本。應該定期對所有重要的資料進行備份。

頻寬

資料傳輸通道最大的傳輸速率。通常是使用網路連線。

BIOS (基本的輸入/輸出系統)

在電腦的電源開啓或重新開機後所啓動的小型程式。它負責起始化硬體元件。大部份的BIOS都允許透過互動式的設定程式來修改低階的系統參數。其程式碼是位於唯讀記憶體(ROM)的晶片中。

書籤(使用瀏覽器)

URL 集中的項目。

開機

電腦從開啓電源到準備好系統供使用的電腦作業程序。

瀏覽器

顯示本機檔案或網頁內容的程式。

用戶端

在網路環境中，連線至伺服器或是要求伺服器資訊的程式或電腦。

指令行

發出指令至電腦的文字模式。

主控台

之前與終端機同義。在Linux中，*虛擬主控台*允許在沒有執行圖形顯示的情況下，讓螢幕用於多個獨立且平行的工作階段。

CPU (中央處理單元)

請參閱處理器。

游標

游標通常是一個區塊或是底線字元，標示出文字輸入的位置。

精靈

精靈(磁碟和執行監視器)是一個在背景執行並在需要時會自動啓動的程式。例如，HTTP精靈(httpd)會回應HTTP的要求。

DDC (直接顯示通道，Direct Display Channel)

在監視器和圖形介面卡之間的通訊標準，允許傳輸某些參數(例如監視器的名稱或解析度)至圖形介面卡。

目錄(在檔案系統中)

包含檔案或是其它目錄(子目錄)的結構。在檔案系統中的目錄為樹狀結構，用以組織檔案。

DNS (領域名稱系統，Domain Name System)

以名稱為基礎的位址與IP 位址之間進行交互轉換的一種協定。

驅動程式

屬於作業系統的一部份，負責硬體元件的通訊。

電子郵件(電子化的郵件)

使用者之間透過網路以電子訊號傳輸郵件的方式。電子郵件地址的格式為username@domain.org。

EIDE (加強型IDE 磁碟介面，Enhanced Integrated Drive Electronics)

改良的IDE 標準，可以讓硬碟空間超過512 MB。

環境

外圍程序所保有的環境變數及其值的集合。使用者可以修改(或是取消設定)現有環境變數的值並設定新的變數。永久指定是透過外圍程序的組態檔案來完成。

環境變數

外圍程序環境中的元素。

乙太網路

電腦網路中資料傳輸的標準。

EXT2 (第二版延伸檔案系統，Second Extended File System)

Linux 支援的檔案系統。

FAQ (常見問題集，Frequently Asked Questions)

是提供「常見問題集」(Frequently Asked Question) 解答之文件的縮寫。

防火牆

過濾網路流量的機制，以保護區域網路免於受到外部未經授權的存取。

FTP (檔案傳輸協定，File Transfer Protocol)

透過以TCP/IP 為基礎的網路來傳輸檔案的協定。

GNOME (GNU 網路物件模型環境，GNU Network Object Model Environment)

Linux 專用的圖形桌面環境。

GNU (GNU is Not Unix)

GNU 是自由軟體基金會(Free Software Foundation, FSF) 的專案。GNU 專案的目標在於建立完整且自由的Unix 式作業系統。「自由」並不是意謂著免費，而是指自由使用：具有取得、修改與變更軟體的權利。現在典型的GNU Manifesto (<http://www.gnu.org/gnu/manifesto.html>) 提供詳細的解釋。從法律的角度來說，GNU 軟體係受GNU General Public License (又稱為GPL)(<http://www.gnu.org/copyleft/gpl.html>) 所保護，也受到GNU Lesser General Public License (又稱為LGPL)(<http://www.gnu.org/copyleft/lgpl.html>) 所保護。適用於GPL 的Linux 核心，可從這個計劃中獲得利益(特別是從工具)，但不應該跟自由混為一談。

GPL (GNU 通用公共授權，GNU General Public License)

請參閱GNU。

主目錄

在檔案系統中的私人目錄，專屬於特定使用者(通常位於/home/<username>)。除了超級使用者root之外，只有該使用者在其主目錄內擁有完整的存取權限。

主機名稱

機器的名稱。通常是在網路上可以找到它的名稱。

HTML (超本文標示語言，Hypertext Markup Language)

「全球資訊網」所使用的文字文件的標記語言。通常會使用瀏覽器來檢視HTML 文件。

HTTP (超本文傳輸協定，Hypertext Transfer Protocol)

定義如何在「全球資訊網」要求和傳輸文件的網路協定。文件通常是由伺服器所提供的HTML 網頁，並由使用者透過瀏覽器來要求。

IDE (整合式磁碟機電子介面)

連接硬碟的標準。

網際網路

以TCP/IP 為基礎的全球電腦網路。

IP 位址

在TCP/IP 網路中某台電腦的唯一(32 位元) 位址。通常以句點分隔的四個十進位來表示(例如，192.168.10.1)

IRQ (中斷要求，Interrupt Request)

(非同步) 要求由硬體或軟體觸發某些動作。大部份的IRQ 是由作業系統所處理。

ISDN (整合服務數位網路，Integrated Services Digital Network)

數位資料透過電話網路傳輸的標準。

KDE (K 桌面環境，K Desktop Environment)

Linux 專用的圖形桌面環境。

核心

核心是作業系統的核心元件。它管理記憶體與檔案系統、包含與硬體設備通訊的驅動程式，並處理程序和網路。

LAN (區域網路，Local Area Network)

LAN 通常是一個相當小的區域網路。

LILO (Linux Loader)

安裝在硬碟開機磁區的小型程式，可以啓動Linux 或另一個作業系統。

連結

連結(在檔案系統中) 指向檔案的指標。它包含硬連結與符號連結。硬連結指到檔案系統中的絕對位置，符號連結指到相對名稱。

Linux

類似Unix 的高效能作業系統核心，可根據GPL (GNU) 自由散佈。這個名稱是一個縮寫(*Linus' uniX*)，而且是依其建立者Linus Torvalds 的名字所訂出來的。雖然這個名稱嚴謹定義僅代表核心本身，但一般人的認知是將*Linux* 代表整個系統。

登入

使用者的認證由使用者名稱和密碼所構成，必須正確才能存取電腦系統或是網路。

登出

關閉互動式Linux 工作階段的程序。

主要記憶體

允許幾乎沒有延遲的隨機存取之實體記憶體。這通常稱為RAM (隨機存取記憶體，Random Access Memory)。

man 頁面

Unix 系統傳統形式的文件，可以使用man 指令來讀取。man 頁面通常是以參考樣式來撰寫。

MBR (主開機記錄)

硬碟的第一個實體磁區，其內容會最先載入到主記憶體中並由BIOS 執行。這個程式碼接著會從硬碟分割區載入作業系統或是更複雜的開機載入器，例如LILO 或GRUB。

MD5

產生雜湊值的演算法(檔案的MD5 檢查總數)。與原始檔案內容不同的檔案所建立的MD5 檢查總數，幾乎不可能產生相同的檢查總數檔案。

裝載

將檔案系統附加到系統目錄樹狀結構的程序。

多工作業

作業系統可以(幾乎)同時執行多個程序的能力。

多重使用者

作業系統可以讓多重使用者在一部電腦上同時執行工作的能力。

網路

數台電腦的交互連接，以允許彼此之間進行資料的傳輸。通常透過網路傳送要求的電腦被稱為用戶端，然而回應要求(例如，傳送文件)的電腦則稱為伺服器。

NFS (網路檔案系統，Network File System)

存取網路檔案系統的協定。

NIS (網路資訊服務，Network Information Service)

網路的中央使用者管理系統。NIS 可以管理整個網路的使用者名稱和密碼。

作業系統

請參閱核心。

分割區

硬碟中的區段，包含檔案系統或是交換空間。

路徑

在檔案系統中描述檔案位置的唯一表示。

隨插即用

自動化的硬體偵測與組態協定。

處理程序

執行中的程式。有時稱為工作。

處理器

處理器(CPU，中央處理單元)是一種微晶片，用以執行儲存在主記憶體中的機器碼。它是電腦的大腦。

提示

在每個指令行開始所列印的簡短(可設定的)字串。它通常包含目前的工作目錄。

協定

對於硬體、軟體或網路的標準定義介面與通訊方法。例如HTTP與FTP協定。

Proxy

通常是指做為中介儲存的電腦，以利從網際網路傳輸資料。如果相同的文件要求的次數超過一次，則第二次要求時可以獲得更快速的服務。欲利用此功能的電腦必須加以設定，才能透過Proxy發出要求。

RAM (隨機存取記憶體，Random Access Memory)

請參閱主要記憶體。

ReiserFS

允許快速修復潛在不一致的檔案系統類型。作業系統關閉前，若未卸載檔案系統，就會發生這種不一致，例如，電源中斷。

root

超級使用者帳戶。超級使用者擁有的所有的權限。此帳戶是用於管理工作，不應該用於一般的工作。

根目錄

在檔案系統階層中的基礎目錄。在UNIX 當中，根目錄以/ 表示。

SCSI (小型電腦系統介面，Small Computer Systems Interface)

連接硬碟以及其他像是掃描器和磁碟等設備的標準。

伺服器

專門用以提供服務的電腦或程式，通常是透過網路。例如HTTP、DNS 以及FTP。

外圍程序

允許發出指令的互動式程式。外圍程序有數個類型，例如Bash、zsh 以及tcsh。外圍程序的每個類型都有其特定的程式語言。

SMTP (簡單郵件傳送通訊協定，Simple Mail Transfer Protocol)

透過網路傳送電子郵件的協定。

SSL (Secure Socket Layer)

傳輸HTTP 資料的加密協定。

超級使用者(superuser)

請參閱root。

系統管理員

請參閱root。

工作

請參閱處理程序。

TCP/IP

用於網際網路的通訊協定，也用於大部份的區域網路。

telnet

Telnet 是和遠端主機通訊的協定。遠端登入的telnet 基本上已由提供加密連線的SSH 所取代。

終端機

以前是連接到中央電腦的鍵盤和螢幕組合。今日這個詞彙大多是指模擬實際終端機的程式(例如xterm)。

Tux

Linux 企鵝的名稱。請參閱<http://www.sjbaker.org/tux/>。

UNIX

UNIX 是一種作業系統，也是註冊商標。

URL (一致資源定址器，Uniform Resource Locator)

由協定(例如，<http://>)、主機與網域的名稱(例如，www.suse.de) 以及文件(例如，[/us/company/index.html](http://www.suse.de/us/company/index.html)) 所組成的網路資源規格。完整 URL 的範例：<http://www.suse.de/us/company/index.html>。

使用者目錄

請參閱主目錄。

VESA (視訊電子標準協會，Video Electronics Standard Association)

定義視訊標準的產業論壇。

萬用字元

單一位置的保留字元(符號為：?) 或多個不明字元的保留字元(符號為：*)。這些屬於一般表示式的一部份。

視窗管理員

在 X Window 系統上執行的程式，它允許像是調整視窗大小或是移動它們的動作。視窗管理員也負責像是視窗標題與框線等視窗裝飾。使用者可以自訂其行為和外觀。

WWW (全球資訊網，World Wide Web)

架構在 HTTP 的通訊協定，這是一個文件、檔案和影像的超連結集合，可以透過網頁瀏覽器來檢視。

X Window 系統

X Window 系統是以視窗系統為基礎的網路，可在廣泛的電腦上執行。它提供像是繪製線條或矩形等工具。它是硬體和視窗管理員之間的中間層。

X11

X Window 系統的第 11 個版本。

YaST (Yet another Setup Tool)

SUSE LINUX 系統助手。

YP

請參閱 NIS。

Index

symbols

.local 做為頂層領域	98
主控台	
- 切換	179
- 圖形化	
· 關閉	76, 159
- 指定	179
主機名稱	51
主開機磁區	see MBR
介面卡	
- 圖形	186
- 廣播	49
- 聲音	48
- 電視	49
位址	
- IP	330
- MAC	330
使用者	
- /etc/passwd	320, 425
- 利用YaST 進行管理	52
來源	
- 編譯	115
保全性	
- 設定	52-56
- tcpd	525
修復系統	119
偽裝	504
- 使用SuSEfirewall2 設定	506
備份	43
- 利用YaST 來建立	56
- 還原	57
冷插拔	298
分割	

- 建立	58
分割區	
- 分割區表格	146
- 加密	514
- 參數	60
- 建立	9, 60
- 調整Windows 的大小	14
- 類型	9
- fstab	62
- LVM	61
- RAID	61
- swap	61
列印	205, 208-211
- 佇列	209
- 應用程式，由	213
- 指令行	213
- 測試頁	210
- 疑難排解	
· 網路	220
- 網路	
· 疑難排解	220
- 設定YaST	208
- 連接埠	209
- 連線	209
- 驅動程式	209
- CUPS	214
- foomatic-filters	95
- GDI 印表機	219
- Ghostscript 驅動程式	209
- IrDA	290
- kprinter	214
- LPRng	95
- PPD 檔案	209

- Samba	473
- xpp	214
加密	
- 分割區	514
- 檔案	514
協定	
- FTP	434
- HTTP	434
- HTTPS	434
- LDAP	411
卡	
- 圖形	
· 驅動程式	196
- 網路	341
· 測試	340
可插式驗證模組	see PAM
可移動的媒體	
- subfs	101
同步化資料	229
- 電子郵件	229
- Evolution	232
- Kontact	232
- KPilot	232
名稱伺服器	see DNS
命令	
- getfacl	531
啓動	
- 圖形	76, 159
· 關閉	76, 159
啓動中	
- GRUB	145
國際化	180
圖形	
- 卡	
· 驅動程式	196
圖形使用者介面	184-192
埠	
- 掃描	494
多重廣播DNS	98
套件	
- 以build 編譯	117
- 套件管理員	109
- 安裝	110
- 建置	95
- 確認	109
- 編譯	115
- 解除安裝	110
- LSB	109
- RPM	109
套件過濾器	see 防火牆

子版本	455
字型	198
- CID-keyed	201
- TrueType	197
- Xft	198
字型	
- X11 核心	201
安全性	516-525
- 加密檔案系統	230
- 問題與	519, 521
- 報告問題	525
- 密碼	518
- 工程	517
- 序列終端機	517
- 序號終端機	517
- 攻擊	522-523
- 本地	518-520
- 病毒	520
- 秘訣與技巧	523
- 網路	520-523
- 蠕蟲	523
- 許可權	519
- 開機	517-518
- 防火牆	55, 502
- DNS	522
- RPM 簽名	524
- Samba	476
- Squid	484
- SSH	510-514
- X 以及	520
安裝	
- 套件	110
- 媒體檢查	44
- 從網路	80
- 文字模式	74-75
- 開機載入器	75
- GRUB	147
- VNC	73
- YaST	3-29
安裝支援	
- 3D 繪圖卡和	204
密碼檔案系統	514
尋找	174
快閃磁碟機	
- 開機自	146
手動安裝	107
指令	
- 熱插拔	295
- chown	98
- fonts-config	198

- free	175
- grub	147
- head	98
- hwinfo	297
- ldapadd	421
- ldapdelete	423
- ldapmodify	423
- ldapsearch	423
- lp	213
- nice	98
- rpm	109
- rpmbuild	109
- scp	511
- setfacl	532
- sftp	511
- smbpasswd	477
- sort	98
- ssh	510
- ssh-agent	513
- ssh-keygen	513
- tail	98
- udev	301
掃描	
- 疑難排解	47
- 設定	46
授權	<i>see</i> GPL
搖桿	
- 設定	192
支援	64
救援系統	122
- 使用	123
- 啓動	123
數位相機	231
數據機	
- 纜線	347
- YaST	343
日誌檔	
- apache2	442, 451
- httpd	441, 442, 451
- Unison	460
- XFree86	203
時區	63
更新	91-95, 117
- 修補程式光碟	41
- 問題	92
- 密碼和群組	92
- 線上	39-41
- YaST	93
服務位置通訊協定	<i>see</i> SLP
本地	

- UTF-8	98
核心	164-169
- 來源	164
- 參數	164
- 安裝	168-169
- 快取	176
- 模組	166-167
- 網路卡	341
- 編譯	168
- modprobe.conf	96
- 模組載入器	167
- 版本2.6	96
- 精靈	167
- 編譯	164, 168
- 設定	164-165
- 錯誤訊息	168
- 限制	315
- kmod	167
- modprobe.conf	167
核心檔	174
檔案	
- 加密	514
- 同步化	453-469
- 子版本	455
- CVS	454, 460-462
- mailsync	455, 467-469
- rsync	455
- Unison	454, 458-460
- 尋找	174
檔案伺服器	51
檔案系統	308-316
- 修復	124
- 加密	514
- 支援的	313-314
- 檔案系統檢查	561
- 詞彙	308
- 選取	308
- 限制	314
- ACL	528-536
- Ext2	309-310
- Ext3	310-311
- FAT	15
- JFS	312
- LFS	314
- NTFS	15, 16
- Reiser4	311-312
- ReiserFS	309
- sysfs	294
- XFS	312-313
權限	

- 檔案權限	174	· boot.setup	138
正在設定		· halt	138
- GRUB	153	· nfsserver	395
滑鼠		· portmap	395
- 設定	192	· rc	136–138
無線連線		· squid	486
- 藍芽	281	- irda	290
熱插拔	293–299	- mkinitrd	133
- 事件	295	- modify_resolvconf	176, 353
- 事件記錄器	298	- SuSEconfig	141–142
- 代辦		筆記型電腦	226–230, <i>see</i> 膝上型電腦
· PCI	297	- 硬體	226
- 代辦程式	295	- 電源管理	226, 251–260
· 介面	295	- PCMCIA	226
· 設備	295	- SCPM	227
· USB	297	- SLP	228
- 儲存設備	296	系統	
- 對應檔案	297	- 停止不動	76
- 日誌檔	298	- 安全性	54
- 模組		- 救援	122
· 自動化載入	297	- 更新	42, 91–95, 117
- 白名單	297	- 當地語系化	180
- 網路設備	296	- 設定	31–65
- 設備名稱	294	- 語言	64
- 錯誤分析	298	- 限制資源使用	174
- 黑名單	297	系統服務	51
- PCI	297	系統監控	228
當地語系化	180	- KPowersave	228
硬碟		- KSysguard	228
- DMA	45	組態	
硬體		- PAM	108
- 硬碟控制器	45	- Squid	488
- 資訊	45	組態檔	351
- CD-ROM	44	- .bashrc	172, 175
- ISDN	344	- .mailsync	467
- SCSI 設備	80	- .profile	172
磁片		- /etc/powersave.conf	105
- 救援	57	- 設定檔	172, 175, 182
- 開機	57	- 語言	180, 182
- 開機自	146	- 輸出	395
磁碟		- crontab	172
- 磁片		- csh.cshrc	182
· 格式化	78	- dhcp	352
- 開機		- dhcpd.conf	400
· 建立	158	- fstab	62
程序檔		- grub.conf	153
- boot.udev	305	- host.conf	354
- init.d	134, 137–140, 357	· multi	354
· 開機	138	- HOSTNAME	357
· boot.local	138	- hosts	51, 340, 353

- httpd.conf 439
- hwinfo 297
- ifcfg-* 352
- inittab 136, 179
- menu.lst 148
- named.conf 371, 374–380
- network 352
- networks 354
- nscd.conf 356
- nsswitch.conf 355, 424
- pam_unix2.conf 424
- resolv.conf 176, 352, 371
- slapd.conf 416
- wireless 352
- xorg.conf 192
 - Device (設備) 196
 - Monitor (顯示器) 197
 - Screen (螢幕) 194

組態檔案

- .emacs 176
- .xsession 513
- 密碼 92
- 服務 476
- 核心 133
- 權限 524
- 熱插拔 294
- 組態 164
- 群組 92
- 路由 352
- 輸出 395, 493
- acpi 254
- apache2 438
- asound.conf 49
- dhclient.conf 399
- foomatic/filter.conf 95
- fstab 123
- gshadow 99
- host.conf
 - 順序 354
 - alert 354
 - nospooft 354
 - trim 354
- httpd.conf 438
- hwup 295
- inittab 134, 135
- inputrc 180
- irda 290
- modprobe.conf 49, 96, 167
- modules.conf 96
- modules.dep 167

- named.conf 487
- powersave 254
- resolv.conf 486
- samba 476
- smb.conf 472, 473
- smppd.conf 358
- smpppd-c.conf 359
- squid.conf ... 486, 488, 490, 493, 495, 497
- squidguard.conf 497
- sshd_config 513
- sysconfig 63, 141–142
- termcap 180
- XF86Config ... *see* 組態檔案, xorg.conf
- xml/catalog 95
- xml/suse-catalog.xml 95
- xorg.conf 108

網路

- 基本的網路位址 332
- 廣播位址 332
- 本地主機 332
- 無線 229
- 網路遮罩 331
- 藍芽 230, 283
- 設定 49–52, *hyperpage*349, 340 – –349
 - IPv6 339
- 路由 52, 330, 331
- DHCP 50, 397
- DNS 339
- IrDA 230
- SLP 361
- TCP/IP 328
- WLAN 230
- YaST 341

網路檔案系統 *see* NFS

網路資訊服務 *see* NIS

網路驗證

- Kerberos 107

網際網路

- 撥號 357–359
- 網頁伺服器 *see* Apache
- cinternet 359
- DSL 347
- ISDN 344
- kinternet 359
- qinternet 359
- smpppd 357–359
- TDSL 349

網頁伺服器

- Apache *see* Apache

線串套件

- NPTL	97	- Firewire (IEEE1394)	231
編碼		- PDA	231
- ISO-8859-1	181	- USB	231
- UTF-8	98	行動電話	231
編輯器		裝置節點	
- Emacs	176-177	- udev	301
- vi	177	解析程式庫	
繪圖		- .local 做為頂層領域	98
- 卡		解除安裝	
· 3D	202-204	- GRUB	158
- 3D	202-204	- Linux	158
· 安裝支援	204	記憶體	
· 支援	202	- RAM	175
· 測試	203	記錄	
· 疑難排解	203	- 登入次數	55
· 診斷	203	- logrotate	
· 驅動程式	202	· 設定	173
· 3Ddiag	203	記錄檔	173
· SaX	203	- 訊息	64, 372, 509
- GLIDE	202-204	- 記錄	55
- OpenGL	202-204	- boot.msg	64
· 測試	203	記錄檔案	
· 驅動程式	202	- boot.msg	254
群組		- Squid	487, 489, 494
- 管理	53	設定	141
聲音		- 使用者	52
- 在YaST 中設定	48	- 安全性	52-56
- 音色	48	- 列印	208-211
膝上型		- 圖形介面卡	186
- IrDA	289-291	- 廣播	49
膝上型電腦		- 掃描器	46
- SCPM	241	- 搖桿	192
藍芽	230, 281	- 數據機	343
- 網路	283	- 時區	63
- hciconfig	285	- 滑鼠	192
- hcitool	285	- 硬碟	
- opd	287	· DMA	45
- pand	286	- 硬碟控制器	45
- sdptool	285	- 硬體	44-49
虛擬控制台		- 系統	31-65
- 切換	63	- 系統服務	51
虛擬記憶體	61	- 網路	49-52, 341
螢幕		· 手動	349
- 解析度	195	- 纜線數據機	347
行動性	225-232	- 群組	53
- 外接硬碟	231	- 膝上型電腦	235-240
- 數位相機	231	- 語言	64
- 筆記型電腦	226	- 路由	52, 352
- 行動電話	231	- 軟體	33-43
- 資料安全性	230	- 鍵盤	192

- 防火牆	55
- 電子郵件	50
- 電視	49
- 音效卡	48
- Apache	438–442
- CD-ROM	44
- DNS	51, 365
- DSL	347
- GRUB	147
- IPv6	339
- IrDA	289
- ISDN	344
- NFS	51
- NTP	
· 用戶端	51
- Samba	473–476
· 伺服器	52
· 用戶端	52, 479
- SSH	510
- T-DSL	349
- X	184
設定檔	
- logrotate.conf	173
許可權	
- ACL	528–536
語言	64
說明	
- info 頁面	174
- man 頁面	174
- X	197
變數	
- 環境	180
資料安全性	230
路由	52, 330, 352
- 偽裝	504
- 網路遮罩	331
- 路由	352
- 靜態	352
軟體	
- 安裝	33–38
- 移除	33–38
- 編譯	115
軟體RAID	<i>see</i> RAID
輕量型目錄存取協定	<i>see</i> LDAP
輸入法	
- CJK	180
通訊協定	
- IPv6	333
- SLP	361
- SMB	472
邏輯磁碟區管理	<i>see</i> LVM
錯誤訊息	
- 權限遭拒	62
- 解譯器不良	62
鍵盤	
- 亞洲字元	180
- 設定	192
- 配置	179
· 組合	180
· 組合鍵	180
- X 鍵盤延伸程式	180
- XKB	180
開機	131
- 從CD 2	79
- 方法	75
- 磁片，從	79
- 管理	146
- 系統停止不動	76
- 記錄	64
- 設定	19
· YaST	155–158
- 載入程式	157
· 位置	157
- 開機磁區	146
- 開機管理員	146
- CD，從	4
- GRUB	76, 147–161
- initrd	
· 建立	133
- LILO	76
- USB 隨身碟	146
開機片	146
- 使用dd 建立	78
- 使用rawrite 建立	77
- 建立	
· DOS	77
- CD	146
防火牆	55, 502
- 套件過濾器	502
- 封包過濾器	505
- Squid 與	493
- SuSEfirewall2	502, 506
隨身碟	231
電子郵件	
- 同步化	229, 455
· mailsync	467–469
- 設定	50
電源管理	226, 251–266
- 休眠	252
- 充電強度	263

- 待命 252
- 暫停 252
- 電池監視器 253
- ACPI 251, 254–259, 262
- APM 251, 253–254, 262
- cpufreqency 260
- cpuspeed 260
- powersave 260
- YaST 267

電視

- 介面卡組態 49

- 電話交換機 345

音效

- 混音器 106

- 領域名稱系統 *see* DNS

- 顯示器設定 184

驗證

- PAM 317–324

- 64 位元Linux 127

- 執行期間支援 128

- 核心規格 130

- 軟體開發 128

A

- ACL 527–536

- 存取 528, 531

- 定義 528

- 支援 536

- 效用 533

- 檢查演算法 535

- 結構 529

- 處理 529

- 許可權位元 530

- 遮罩 532

- 預設值 528, 533

ACPI

- 關閉 5

- Apache 51, 433–452

- 安全性 52–56

- 內容協商 436

- 啓動 437

- 執行緒 436

- 安全性 450–451

- 安裝 437–438

- 旗標 439

- 模組 435

- 啓動 438

- 載入 439

- mod_perl 445

- mod_php4 447

- mod_python 447

- mod_ruby 447

- 疑難排解 451

- 虛擬主機 436, 447–450

- 記錄 441, 442

- 設定 438–442

- 許可權 440, 450

- 預設網頁 435

- apxs 438

- CGI 444

- DocumentRoot 439

- Squid 495

- SSI 442, 443

B

Bash

- .bashrc 172

- .profile 172

- 設定檔 172

- BIND 370–380

BIOS

- 病毒防護 76

- 開機順序 4

- booting 561, 565, 569, 573

C

CD

- 開機自 4, 146

CD-ROM 光碟機

- 支援的 80

- chown 98

- CJK 180

commands

- e2fsck 565

- jfs_fsck 573

- slptool 363

- xfs_check 569

configuration files

- suseconfig 142

- cpuspeed 260

- crashes 561, 565, 569, 573

- cron 172

- CVS 454, 460–462

D

- deltarpm 112

- depmod 167

- DHCP 397–404

- 伺服器 400–403

- 使用YaST 設定 398

- 套件	399
- 靜態位址指定	403
- dhcpd	400–403
DNS	50, 339
- 區域	
· 檔案	378
- 反向查詢	380
- 名稱伺服器	352
- 啓動	372
- 安全性與	522
- 最上層領域	340
- 疑難排解	372
- 記錄	376
- 設定	51, 365
- 轉寄	372
- 選項	375
- 郵件交換器	340
- 領域	352
- BIND	370–380
- NIC	340
- Squid 與	487
DOS	
- 共用檔案	471
E	
e2fsck	565
Emacs	176–177
- .emacs	176
- default.el	176
Evolution	232
F	
FAT 檔案系統	15
file systems	
- e2fsck	565
- jfs_fsck	573
- reiserfsck	561
- xfs_check	569
Firewire (IEEE1394)	
- 硬碟	231
G	
GPL	577
GRUB	145–161
- 主開機記錄(MBR)	146
- 分割區名稱	149
- 功能表編輯器	151
- 啓動密碼	154
- 指令	147–155
- 疑難排解	160

- 萬用字元	152
- 解除安裝	158
- 設備名稱	149
- 開機	147
- 開機功能表	148
- 開機磁區	146
- 開機管理	146
- 限制	147
- device.map	147, 153
- GRUB 位置錯誤	160
- GRUB 外圍程序	154
- grub.conf	147, 153
- JFS 和GRUB	160
- menu.lst	147, 148

H

hciconfig	285
hcitool	285
head	98
hwinfo	297

I

I18N	180
inetd	51, 94
info 頁面	174
init	134–135
- 新增程序檔	139
- 程序檔	137–140
- inittab	134
insmod	167
IP 位址	
- 偽裝	504
- 動態指定	397
- 私人	332
- 類別	331
- IPv6	333
· 設定	339
IrDA	230, 289–291
- 停止	289
- 啓動	289
- 疑難排解	290
- 設定	289

J

jade	<i>see</i> SGML, openjade
jade_dsl	94
jfs_fsck	573–575

K

Kmod	<i>see</i> 核心，模組載入器
------	---------------------

Kontakt	232
KPilot	232
KPowersave	228
KSysguard	228

L

L10N	180
LDAP	51, 411-431
- 伺服器組態	416
- 修改資料	422
- 刪除資料	423
- 存取控制	419
- 搜尋資料	423
- 新增資料	420
- 目錄樹	413
- 管理使用者	429
- 管理群組	429
- ACL	417
- ldapadd	420
- ldapdelete	423
- ldapmodify	422
- ldapsearch	423
- YaST	
- 模組	425
- 樣板	425
- YaST LDAP 用戶端	424
LFS	314
LILO	
- 設定	76
Linux	
- 網路和	327
- 與另一個OS 共用檔案	471
- 解除安裝	158
linuxrc	72
- 手動安裝	107
linuxthreads	97
logrotate	173
LSB	
- 安裝套件	109
lsmod	167
LVM	
- YaST	81

M

man 頁面	174
MBR	146
modinfo	167
modprobe	167
mountd	395
Mozilla	

- 安全性	52
-------------	----

N

NAT	<i>see</i> 偽裝
NetBIOS	472
networks	
- 組態檔	351-357
NFS	391
- 伺服器	51, 393
- 權限	395
- 用戶端	51, 392
- 裝載	392
- 輸入	392
- 輸出	394
nfsd	395
NGPT	97
nice	98
NIS	51, 385-389
- 主要	386-388
- 次要	386-388
- 用戶端	388
NPTL	97
NSS	355
- 資料庫	355
NTFS 檔案系統	15
NTP	
- 用戶端	51
nVidia	94

O

opd	287
OpenSSH	<i>see</i> SSH
OS/2	
- 共用檔案	471

P

PAM	317-324
- 組態	108
pand	286
PCMCIA	226, 234
- 公用程式	237
- 卡片管理員	234
- 數據機	236
- 疑難排解	237
- 網路卡	236
- 設定	235
- IrDA	289-291
- ISDN	236
- SCSI	236
PDA	231

ports	
- 53	375
PostgreSQL	
- 更新	93
powersave	260
- 設定	261
proxies	
- 快取	483
Proxy	52
proxy	<i>see Squid</i>
- 優點	483
- 操作順暢	492

R

RAID	
- YaST	86
reiserfsck	561
RFC	328
rmmod	167
RPM	109–117
- 依存度	110
- 修補程式	111
- 安全性	524
- 工具	117
- 更新	110
- 查詢	113
- 版本	95
- 確認	109
- 解除安裝	110
- 資料庫	
· 重建	110, 115
- 驗證	114
- deltarpm	112
- rpmnew	110
- rpmorig	110
- rpmsave	110
- SRPMS	116
rpmbuild	95, 109
rsync	455, 465
runlevel	63, 135–137
- 切換	63
- 變更	136–137
runlevels	
- 在YaST 中編輯	140

S

Samba	471–481
- 伺服器	52, 473–476
- 停止	473
- 共用	473, 474

- 列印	480
- 印表機	473
- 名稱	473
- 啓動	473
- 安全性	476
- 安裝	473
- 最佳化	480
- 權限	476
- 用戶端	52, 473, 479–480
- 登入	476
- 設定	473–476
- 說明	481
- SMB	472
- swat	476
- TCP/IP 以及	472
SaX	184
- 多重顯示器	189
SCPM	63, 241
- 切換設定檔	244
- 啓動	243
- 筆記型電腦	227
- 管理設定檔	243
- 資源群組	243
- 進階設定	244
scripts	
- init.d	
· network	357
· nfsserver	357
· portmap	357
· sendmail	357
· xinetd	357
· ypbind	357
· ypserv	357
- SuSEconfig	
· disabling	142
SCSI 設備	
- 檔案名稱，指定	80
- 設定	80
sdptool	285
SGML	
- 目錄	101
- openjade	94
SLP	228, 361
- 瀏覽器	363
- 註冊服務	362
- Konqueror	363
- slptool	363
SMB	<i>see Samba</i>
sort	98
spm	115

Squid	483	- 層模型	328
- 停止	487	- ICMP	328
- 功能	483	- IGMP	328
- 啓動	486	- TCP	328
- 報告	498	- UDP	328
- 存取控制	495		
- 安全性	484	U	
- 快取	483, 484	udev	301
大小	485	- 啓動程序檔	305
- 操作順暢的Proxy	492, 494	- 大容量儲存	304
- 權限	486, 490	- 硬碟	305
- 物件狀態	485	- 自動化	303
- 疑難排解	487	- 萬用字元	303
- 目錄	486	- 規則	302
- 系統需求	485	- 關鍵字	303
- 組態	488	- sysfs	304
- 統計資料	495, 496	- udevinfo	304
- 解除安裝	487	ulimit	174
- 記錄檔案	487, 489, 494	- 選項	174
- 防火牆與	493	updatibg	
- ACL	490	- 混音器	106
- Apache	495	USB	
- cachemgr.cgi	495, 496	- 硬碟	231
- Calamaris	498	- 隨身碟	231
- CPU 與	486	UTF-8	
- DNS	487	- 編碼	98
- RAM 與	486		
- squidGuard	496	V	
SSH	510-514	VNC	
- 精靈	512	- 安裝	73
- 金鑰組合	512, 513	- 管理	52
- 驗證機制	513		
- scp	511	W	
- sftp	511	whois	340
- ssh	510	Windows	
- ssh-agent	513	- 共用檔案	471
- ssh-keygen	513	WLAN	230
- sshd	512		
- X 與	513	X	
subfs		X	183
- 可移動的媒體	101	- 多重顯示器	189
subversion	462	- 字型	197
SUSE LINUX		- 字型系統	198
- 安裝	72	- 字集	197
sx	95	- 安全性	520
		- 最佳化	192-197
T		- 虛擬螢幕	195
tail	98	- 設定	184
TCP/IP	328	- 說明	197
- 封包	329	- 驅動程式	196

- 3D	188	- 安裝建議	7
- CID-keyed 字型	201	- 安裝模式	7
- SaX2	193	- 安裝範圍	17
- SSH 與	513	- 廣播卡	49
- TrueType 字型	197	- 建立磁片	57
- X11 核心字型	201	- 掃描器	46
- xf86config	193	- 控制中心	32
- Xft	198	- 搖桿	192
- xft	197	- 支援要求	64
X 鍵盤延伸程式	<i>see</i> 鍵盤、X 鍵盤延伸程式	- 數據機	343
X Window System	<i>see</i> X	- 文字模式	65–70, 74–75
X.Org	192	- 模組	68
XF86Config		- 疑難排解	75
- Modeline (模式行)	195	- 時區	63
xfs_check	569	- 更新	41, 42, 93
Xft	198	- 根密碼	21
xinetd	94	- 滑鼠	8, 192
XKB	<i>see</i> 鍵盤、X 鍵盤延伸程式	- 硬碟控制器	45
XML		- 硬體	44–49
- 目錄	95, 101	- 硬體資訊	45
- openjade	94	- 磁碟分割	9
xorg.conf		- 磁碟空間	10
- color depth (色彩深度)	195	- 系統啓動	4
- Depth (深度)	195	- 系統安全性	54
- Device (設備)	195	- 網路卡	341
- Display (顯示)	195	- 網路組態	22, 49–52
- Files (檔案)	193	- 線上更新	39–41, 68
- InputDevice (輸入設備)	193	- 纜線數據機	347
- modelines (模式行)	194	- 群組管理	53
- Modes (模式)	194, 195	- 設定	31–65
- Monitor (顯示器)	194, 195	- 設定檔管理員	63
- ServerFlags (伺服器標誌)	193	- 語言	64
		- 語言選擇	7, 32
Y		- 路由	52
YaST		- 軟體	33–43
- 主機名稱	51	- 軟體更新	23
- 使用者管理	52	- 鍵盤配置	8, 192
- 修復系統	119	- 開機模式	19
- 備份	43, 56	- 開機組態	155
- 分割	58	- 防火牆	55
- 列印	208–211	- 電子郵件	50
- 啓動	4, 32	- 電源管理	267
- 圖形介面卡	184, 186	- 電視卡	49
- 圖形使用者介面	184–192	- 音效卡	48
- 套件相依性	18	- 顯示器設定	184
- 套件管理員	34	- 驅動程式光碟	65
- 媒體檢查	44	- 3D	202
- 安全設定	6	- CD-ROM	44
- 安裝使用	3–29	- DHCP	398
- 安裝來源	39	- DMA	45

- DNS	51
- DSL	347
- ISDN	344
- LDAP 用戶端	424
- LVM	58, 81
- ncurses	65
- NFS 伺服器	51
- NFS 用戶端	51
- NIS 用戶端	25, 388
- NTP · 用戶端	51
- RAID	86

- runlevels	140
- Samba · 伺服器	52
· 用戶端	52, 479
- SCPM	63
- sendmail	50
- SLP 瀏覽器	363
- sysconfig 編輯器	63, 143
- T-DSL	349
- YOU	39–41
YP	<i>see</i> NIS