文 档 信 息

文件状态： [√] 草稿 [ ] 正式发布 [ ] 正在修改	文件标识：	RHEL 入门
	文件位置：	
	当前版本：	1.0
	作者：	panda
	发布日期：	2006/8/26

文 档 更 改 记 录

版本	更改日期	更改人	更改原因	说明
0.1	2006/8/14	潘登	创建	
0.11	2006/8/15	潘登	添加内核,引导部分	
0.12	2006/8/16	潘登	添加磁盘管理,软件安装管理,基础命令	
0.13	2006/8/18	潘登	添加网络部分网络命令,以及 DHCP 和 DNS 服务	
0.14	2006/8/21	潘登	添加 telnet,ssh,vnc,以及 Apache,Squid,Sendmail,Postfix, Cyrus-SASL,dovecot	
1.0	2006/8/26	潘登	添加剩余部分,架构完成	

【目录】

☆ 23
1 ☆Linux正确读音及音标 23
2 ☆Linux相关网站 23
3 ☆Linux安装 24
4 ☆Linux网络安装 26
5 ☆Linux启动盘 28
6 ☆PXE 28
7 ☆PXE安装 29
8 ☆system-config-netboot 33
9 ☆Kickstart 33
10 ☆ks.cfg 33
11 ☆测试kickstart 36
12 ☆Linux启动顺序 36
13 ☆MBR 37
14 ☆OS Loader 37
15 ☆GRUB 37
16 ☆GRUB安装 38
17 ☆测试GRUB 38
18 ☆GRUB引导软盘 39
19 ☆GRUB硬盘引导 40
20 ☆GRUB网络引导 41
21 ☆不同操作系统引导代码 42
22 ☆menu.lst 43
23 ☆手工GRUB引导系统 46
24 ☆GRUB的help 46
25 ☆GRUB命令索引 46
26 ☆GRUB加口令 62
27 ☆GRUB中的cat 63
28 ☆GRUB引导Windows 63
29 ☆从DOS引导linux 65
30 ☆GRUB启动FreeBSD 65
31 ☆GRUB命令行模式 65
32 ☆GRUB菜单模式 66
33 ☆GRUB菜单隐藏的解除 66
34 ☆GRUB丢失或损坏的策略 67
35 ☆linux的启动 68
36 ☆initrd 68
37 ☆inittab 70
38 ☆init 75
39 ☆chkconfig 76

40 ☆检查运行服务 77
41 ☆系统服务详解 77
42 ☆登录过程 88
43 ☆issue 89
44 ☆motd 89
45 ☆Linux内核 89
46 ☆为什么需要编译内核和管理内核 90
47 ☆lsmod 90
48 ☆modinfo 90
49 ☆rmmod 90
50 ☆insmod 90
51 ☆modprobe 91
52 ☆depmod 91
53 ☆配置内核需要的软件包 92
54 ☆modprobe.conf 92
55 ☆内核的硬件驱动模块 94
56 ☆编译内核的硬件驱动 94
57 ☆内核的解压和安装 94
58 ☆内核的配置 95
59 ☆内核的编译 96
60 ☆内核的安装 96
61 ☆内核的镜像文件 96
62 ☆mkinitrd 96
63 ☆内核的检验 97
64 ☆内核的清理 97
65 ☆修改GRUB 97
66 ☆内核编译总结 97
67 ☆内核的配置菜单 98
68 ☆内核的配置菜单案例 108
69 ☆Makefile 111
70 ☆.config 114
71 ☆单独编译一个模块 144
72 ☆内核补丁 145
73 ☆sysctl 145
74 ☆sysctl.conf 149
75 ☆Linux单用户模式 150
76 ☆GRUB下单用户登录 150
77 ☆LILO下单用户登录 151
78 ☆软盘下单用户登录 151
79 ☆防止进入单用户 151
80 ☆Linux修复模式 152
81 ☆跨越控制台登录 153

82 ☆GRUB跨越控制台登录 153
83 ☆LILO跨越控制台登录 153
84 ☆linux的livecd光盘或第三方linux系统下跨越控制台登录 154
85 ☆chroot 156
86 ☆chroot安装操作系统 156
87 ☆硬盘结构 159
88 ☆Linux存储设备表示方法 160
89 ☆fdsik 162
90 ☆parted 164
91 ☆sfdisk 166
92 ☆partx 166
93 ☆mkfs 166
94 ☆mkswap 167
95 ☆e2label 168
96 ☆tune2fs 168
97 ☆reiserfstune 169
98 ☆fsck 169
99 ☆mount 170
100 ☆umount 172
101 ☆fstab 172
102 ☆df 173
103 ☆LVM 175
104 ☆LVM建立PV 176
105 ☆LVM建立VG 177
106 ☆LVM激活VG 177
107 ☆LVM移除VG 177
108 ☆LVM为VG增加PV 177
109 ☆LVM从VG移除PV 178
110 ☆LVM在PV间转移数据 178
111 ☆LVM创建LV 178
112 ☆LVM删除LV 179
113 ☆LVM扩展LV 179
114 ☆LVM缩小LV 181
115 ☆e2fsadm 181
116 ☆LVM的快照 182
117 ☆LVM更换卷组硬盘 183
118 ☆LVM迁移卷组 183
119 ☆LVM分割卷组 184
120 ☆LVM转变根文件系统为LVM 186
121 ☆LVM共享LVM卷 188
122 ☆LVM系统启动/关闭 188
123 ☆LVM磁盘分配策略 189

124 ☆lvm实验 189
125 ☆mdadm 191
126 ☆建立RAID 194
127 ☆查看RAID 195
128 ☆挂载RAID 196
129 ☆RAID设置热备盘 197
130 ☆RAID阵列转移 197
131 ☆删除RAID 198
132 ☆RAID替换磁盘(在线方式) 199
133 ☆mdadm.conf 199
134 ☆mkraid 199
135 ☆dd 199
136 ☆fdformat 201
137 ☆mformat 202
138 ☆mkdosfs 203
139 ☆badblocks 204
140 ☆autofs 204
141 ☆auto.master 205
142 ☆auto.misc 205
143 ☆启动autofs 206
144 ☆测试autofs 206
145 ☆Linux配额 206
146 ☆QUOTA使用前配置 206
147 ☆quotacheck 207
148 ☆quotaon 207
149 ☆quotaoff 207
150 ☆edquota 208
151 ☆setquota 209
152 ☆quota 209
153 ☆repquota 209
154 ☆quota实验 210
155 ☆cdrecorder 211
156 ☆dvdrecorder 211
157 ☆mkisofs 211
158 ☆mknod 212
159 ☆mknod实验 212
160 ☆主设备号和次设备号 214
161 ☆devfs 215
162 ☆COM口符号 219
163 ☆kudzu 219
164 ☆lspci 219
165 ☆dmesg 220

166 ☆hwbrowser 221
167 ☆lshal和hal-device-manager 221
168 ☆硬件配置文件和工具 221
169 ☆各重要命令与文件的RPM包出处表 222
170 ☆system-config-packages 222
171 ☆RPM 222
172 ☆RPM安装 223
173 ☆RPM删除 225
174 ☆RPM升级 226
175 ☆RPM查询 227
176 ☆RPM导入签名 230
177 ☆RPM校验已安装的软件包 230
178 ☆RPM校验软件包的完整性 231
179 ☆RPM其它RPM选项 232
180 ☆RPM管理包管理器支持网络安装和查询 232
181 ☆从RPM软件包抽取文件 232
182 ☆RPM的配置文件 233
183 ☆编译源代码包的条件 234
184 ☆安装源代码包 234
185 ☆src.rpm软件包 235
186 ☆以bin结尾的安装包 236
187 ☆基于perl和python的程序的安装 236
188 ☆脚本型安装程序 236
189 ☆源码目录 236
190 ☆文件系统权限 237
191 ☆权限实验 239
192 ☆linux常见目录 241
193 ☆Linux的特殊块设备 244
194 ☆/proc 245
195 ☆/proc/partitions 246
196 ☆/etc目录文件介绍 247
197 ☆Linux目录结构介绍 261
198 ☆Linux的基本配置文件 262
199 ☆Linux下的主要文件 264
200 ☆linux路径 267
201 ☆虚拟控制台 267
202 ☆type 267
203 ☆whereis 268
204 ☆which 268
205 ☆info 268
206 ☆whatis 269
207 ☆apropos 269

208 ☆man 269
209 ☆pwd 269
210 ☆ls 269
211 ☆du 271
212 ☆mkdir 272
213 ☆rmdir 273
214 ☆cd 273
215 ☆touch 274
216 ☆cp 275
217 ☆mv 277
218 ☆rm 278
219 ☆ln 279
220 ☆chmod 281
221 ☆chown 283
222 ☆chgrp 284
223 ☆umask 285
224 ☆lsattr 285
225 ☆chattr 286
226 ☆stat 288
227 ☆file 288
228 ☆cat 288
229 ☆tac 290
230 ☆more 290
231 ☆less 292
232 ☆head 294
233 ☆tail 294
234 ☆od 295
235 ☆cut 295
236 ☆sort 296
237 ☆uniq 298
238 ☆split 298
239 ☆tr 300
240 ☆wc 300
241 ☆aspell 301
242 ☆comm 301
243 ☆diff 302
244 ☆find 303
245 ☆xargs 308
246 ☆locate 308
247 ☆slocate 309
248 ☆updatedb 310
249 ☆grep 310

250 ☆tar 315
251 ☆cpio 318
252 ☆dump 322
253 ☆restore 324
254 ☆dump/restore恢复单个文件 325
255 ☆备份恢复实例 325
256 ☆gzip 331
257 ☆gunzip 332
258 ☆bzip2 332
259 ☆bunzip2 332
260 ☆compress 332
261 ☆uncompress 333
262 ☆zip 334
263 ☆unzip 334
264 ☆rar 334
265 ☆unrar 335
266 ☆zgrep 335
267 ☆用户和组 335
268 ☆UID 336
269 ☆GID 336
270 ☆passwd 337
271 ☆shadow 337
272 ☆group 338
273 ☆gshadow 338
274 ☆skel 339
275 ☆login.defs 340
276 ☆newusers 341
277 ☆chpasswd 341
278 ☆login 342
279 ☆useradd 342
280 ☆passwd 343
281 ☆chage 344
282 ☆pwconv 344
283 ☆pwck 344
284 ☆pwunconv 345
285 ☆useradd 345
286 ☆useradd配置文件 346
287 ☆usermod 347
288 ☆userinfo 348
289 ☆userdel 348
290 ☆userconf 348
291 ☆groupadd 349

292 ☆newgrp 349
293 ☆gpasswd 349
294 ☆grpconv 350
295 ☆grpck 350
296 ☆grpunconv 350
297 ☆groupmod 350
298 ☆groupdel 350
299 ☆id 351
300 ☆groups 351
301 ☆users 351
302 ☆who 352
303 ☆whoami 354
304 ☆whois 354
305 ☆w 354
306 ☆logname 355
307 ☆lastb 355
308 ☆last 356
309 ☆finger 357
310 ☆chfn 358
311 ☆chsh 359
312 ☆su 359
313 ☆sudo 360
314 ☆visudo 362
315 ☆sudoedit 369
316 ☆sliplogin 369
317 ☆logout 369
318 ☆vlock 369
319 ☆sync 370
320 ☆shutdown 370
321 ☆halt 371
322 ☆reboot 372
323 ☆init 372
324 ☆uname 372
325 ☆uptime 373
326 ☆free 373
327 ☆logrotate 374
328 ☆作业 374
329 ☆程序 374
330 ☆进程 375
331 ☆& 375
332 ☆jobs 376
333 ☆nohup 376

334 ☆screen 376
335 ☆at 378
336 ☆atq 380
337 ☆atrm 380
338 ☆batch 380
339 ☆crontab 380
340 ☆anacron 383
341 ☆ps 383
342 ☆pgrep 388
343 ☆kill 388
344 ☆killall 389
345 ☆pkill 390
346 ☆xkill 390
347 ☆skill 390
348 ☆top 392
349 ☆nice 394
350 ☆renice 395
351 ☆pstree 395
352 ☆gitps 397
353 ☆tload 398
354 ☆swatch 398
355 ☆suspend 399
356 ☆sysstat 399
357 ☆expr 405
358 ☆clear 406
359 ☆cal 406
360 ☆date 410
361 ☆sleep 412
362 ☆time 412
363 ☆reset,tset 415
364 ☆mtools 416
365 ☆打印 417
366 ☆system-config-printer 418
367 ☆lp 418
368 ☆lpd 418
369 ☆lpq 418
370 ☆lpr 419
371 ☆enscript 420
372 ☆cancel 420
373 ☆lprm 420
374 ☆troff 421
375 ☆lpc 421

376 ☆lpadmin 422
377 ☆lpstat 422
378 ☆CUPS 422
379 ☆启动CUPS 423
380 ☆cupsd.conf 423
381 ☆client.conf 427
382 ☆CUPS日志 427
383 ☆inetd.conf 427
384 ☆services 428
385 ☆securetty 429
386 ☆mesg 429
387 ☆wall 430
388 ☆write 430
389 ☆talk 431
390 ☆rlogin 432
391 ☆rwho 432
392 ☆fwhois 432
393 ☆rsh 433
394 ☆ftp 433
395 ☆lftp 436
396 ☆gftp 439
397 ☆rcp 440
398 ☆wget 440
399 ☆rsync 445
400 ☆md5sum 447
401 ☆gpg 447
402 ☆gpg实验 449
403 ☆文本编辑器 451
404 ☆vi 452
405 ☆vi参数 452
406 ☆vi命令模式 452
407 ☆vi命令语法 452
408 ☆vi缓冲控制 453
409 ☆vi移动 453
410 ☆vi搜索 454
411 ☆vi文本替换 454
412 ☆vi编辑控制 454
413 ☆vi可视模式 455
414 ☆Linux Shell 456
415 ☆Terminals 456
416 ☆xterms 456
417 ☆Shells 457

418 ☆shell特性 457
419 ☆自动补齐 457
420 ☆命令行的历史记录 457
421 ☆编辑命令行 458
422 ☆命令的排列 458
423 ☆命令的任务调度 458
424 ☆命令的替换 459
425 ☆文件名匹配 459
426 ☆管道 462
427 ☆重定向 462
428 ☆bash配置文件 463
429 ☆bashrc 463
430 ☆profile 463
431 ☆profile.d 463
432 ☆.bash_history 464
433 ☆.bash_logout 464
434 ☆.bash_profile 464
435 ☆.bashrc 464
436 ☆提示符 464
437 ☆$PATH 467
438 ☆命令的别名 468
439 ☆Shell函数 469
440 ☆locale 470
441 ☆export 470
442 ☆source 470
443 ☆history 470
444 ☆alias 471
445 ☆unalias 471
446 ☆echo 472
447 ☆set 472
448 ☆unset 473
449 ☆setenv 473
450 ☆unsetenv 473
451 ☆规则表达式 473
452 ☆sed 474
453 ☆awk 478
454 ☆awk文件语法 479
455 ☆Xorg 491
456 ☆X 491
457 ☆安装x11/xorg 491
458 ☆xorg.conf 491
459 ☆Xorg -configure 494

460 ☆xorgconfig 494
461 ☆startx 494
462 ☆设置分辨率 495
463 ☆配置键盘 495
464 ☆配置鼠标 496
465 ☆ddcprobe 496
466 ☆switchdesk 497
467 ☆prefdm 497
468 ☆gdmsetup 497
469 ☆Xorg.0.log 497
470 ☆setup 497
471 ☆网络接口(interface) 497
472 ☆ifconfig 497
473 ☆ifdown 499
474 ☆ifup 499
475 ☆ifcfg-ethN 499
476 ☆network 500
477 ☆hosts 500
478 ☆resolv.conf 501
479 ☆host.conf 501
480 ☆system-config-network 502
481 ☆netconfig 502
482 ☆adsl-setup 502
483 ☆wvdial 503
484 ☆miitool 503
485 ☆ethtool 504
486 ☆netstat 506
487 ☆arp 508
488 ☆ping 508
489 ☆traceroute 509
490 ☆tcpdump 509
491 ☆route 510
492 ☆DHCP 511
493 ☆DHCP服务工作原理 511
494 ☆DHCP的安装 513
495 ☆DHCP启动停止 513
496 ☆dhcpd.conf 513
497 ☆设置IP作用域 515
498 ☆设置客户端的IP选项 516
499 ☆设置租约期限 517
500 ☆保留特定的IP地址 517
501 ☆分配多网段的lP地址 518

502 ☆DHCP中继代理 519
503 ☆DHCP客户端的配置 520
504 ☆DNS 520
505 ☆DNS查询的工作原理 521
506 ☆Bind的简介 523
507 ☆DNS服务的安装 523
508 ☆named 523
509 ☆DNS防火墙设置 524
510 ☆主配置文件 524
511 ☆主要域名服务器 525
512 ☆设置根区域 525
513 ☆设置正向解析区域 526
514 ☆设置反向解析区域 527
515 ☆辅助域名服务器 528
516 ☆配置仅有缓存的服务器 529
517 ☆DNS区域数据库文件 531
518 ☆根服务器信息文件named.ca 532
519 ☆正向解析区域文件 533
520 ☆反向解析区域文件 537
521 ☆实现负载均衡功能 539
522 ☆实现直接解析域名 539
523 ☆实现泛域名的解析 539
524 ☆基于数据库的二级域名管理 540
525 ☆DNS客户端的配置 540
526 ☆DNS重启 540
527 ☆DNS测试 540
528 ☆nslookup 541
529 ☆DNS安全访问控制 543
530 ☆dnssec-keygen 546
531 ☆DNS的DHCP设置 546
532 ☆telnet 547
533 ☆telnet安装 547
534 ☆telnet启动停止 547
535 ☆telnet防火墙 548
536 ☆telnet配置 548
537 ☆telnet端口 548
538 ☆使用telnet 548
539 ☆telnet用root登录 548
540 ☆SSH 548
541 ☆SSH版本 549
542 ☆SSH安装 550
543 ☆SSH生成密钥 550

544 ☆sshd_config 551
545 ☆SSH的防火墙 552
546 ☆启动停止SSH服务器 552
547 ☆使用SSH 552
548 ☆SSH测试 552
549 ☆SSH问题 552
550 ☆SSH日志文件 553
551 ☆SSH远程登陆图形窗口 553
552 ☆SSH传送文件 553
553 ☆sftp 554
554 ☆scp 555
555 ☆文件传输脚本 555
556 ☆ssh-agent 557
557 ☆xhost 557
558 ☆VNC 558
559 ☆VNC安装 558
560 ☆VNC server密码 558
561 ☆VNC server启动 559
562 ☆vncservers 559
563 ☆vnc防火墙 559
564 ☆vncviewer 559
565 ☆xstartup 559
566 ☆通过浏览器使用vnc 560
567 ☆x0vncserver 560
568 ☆Apache启动 560
569 ☆Apache防火墙 560
570 ☆Apache配置文件 560
571 ☆httpd.conf 561
572 ☆虚拟目录 586
573 ☆主目录浏览 586
574 ☆Apache虚拟主机 587
575 ☆Apache的CGI环境变量 589
576 ☆Apache的CGI配置 590
577 ☆Apache的PHP配置 591
578 ☆apachectl 592
579 ☆Apache的模块功能 592
580 ☆Apache的代理服务器 595
581 ☆Apache配置代理服务器 596
582 ☆Apache配置逆向代理服务器 597
583 ☆Apache配置缓冲服务器 598
584 ☆Apache镜像站点 598
585 ☆htpasswd 599

586 ☆Apache的安全机制 599
587 ☆Apache目录授权 602
588 ☆Apache的用户个人主页 602
589 ☆HTTPS 602
590 ☆Apache查看状态 603
591 ☆Squid 604
592 ☆Squid启动停止 604
593 ☆Squid防火墙 604
594 ☆Squid配置文件 604
595 ☆Squid配置 604
596 ☆Squid配置透明代理 605
597 ☆Squid访问控制列表 606
598 ☆Squid访问控制列表使用MAC 609
599 ☆Squid身份认证 610
600 ☆Squid初始化 611
601 ☆Squid设置错误提示为中文 611
602 ☆Squid日志 611
603 ☆squid实验 611
604 ☆邮件系统 612
605 ☆邮件服务器 613
606 ☆SMTP协议 613
607 ☆POP3 协议 614
608 ☆IMAP4 协议 614
609 ☆POP和IMAP差异 614
610 ☆mail 614
611 ☆uuencode/uudecode 615
612 ☆system-switch-mail 615
613 ☆sendmail 615
614 ☆sendmail安装 615
615 ☆sendmail使用前配置 616
616 ☆sendmail防火墙 618
617 ☆sendmail启动 618
618 ☆sendmail校验 618
619 ☆sendmail收取邮件 620
620 ☆sendmail别名 621
621 ☆sendmail允许转发 621
622 ☆sendmail不允许转发 622
623 ☆sendmail选择性的转发 623
624 ☆sendmail配置文件 623
625 ☆m4 624
626 ☆mc语法 624
627 ☆sendmail.mc 629

628 ☆sendmail.cf 629
629 ☆access 630
630 ☆access.db 630
631 ☆aliases 631
632 ☆.forward 632
633 ☆local-host-names 633
634 ☆trusted-users 633
635 ☆virtusertable 633
636 ☆virtusertable.db 634
637 ☆mailertable 634
638 ☆spamassassin 635
639 ☆statistics 635
640 ☆submit.mc 635
641 ☆Makefile 635
642 ☆sendmail总结 635
643 ☆新建E-Mail帐号 636
644 ☆sendmail邮件限额 636
645 ☆mailstats 636
646 ☆mailq 637
647 ☆sendmail -q 637
648 ☆sendmail管理邮件队列 637
649 ☆SMTP协议的基本命令 638
650 ☆sendmail测试 639
651 ☆postfix安装 640
652 ☆postfix收取邮件 640
653 ☆Postfix启动和校验 640
654 ☆Postfix的别名 641
655 ☆postfix允许转发 642
656 ☆postfix不允许转发 643
657 ☆postfix选择性的转发 643
658 ☆postfix接收和转发的条件 643
659 ☆main.cf 643
660 ☆virtual 646
661 ☆master.cf 646
662 ☆pcre_table 646
663 ☆postfix-files 646
664 ☆postfix-script 646
665 ☆regexp_table 646
666 ☆relocated 646
667 ☆transport 646
668 ☆postconf 647
669 ☆重启postfix 647

670 ☆SMTP认证的配置 647
671 ☆Cyrus-SASL 647
672 ☆Cyrus-SASL安装 648
673 ☆Cyrus-SASL启动 648
674 ☆Cyrus-SASL认证机制 648
675 ☆saslauthd配置 648
676 ☆测试Cyrus-SASL 648
677 ☆Cyrus-SASL配置MTA 649
678 ☆dovecot配置 650
679 ☆dovecot.conf 651
680 ☆mail日志 663
681 ☆Mail测试 663
682 ☆测试pop/imap 664
683 ☆procmail 666
684 ☆使用POP客户端 667
685 ☆fetchmail 668
686 ☆squirrelmail 671
687 ☆SMB协议 671
688 ☆Samba 672
689 ☆安装samba 673
690 ☆Samba使用前配置 673
691 ☆Samba防火墙 674
692 ☆samba启动 674
693 ☆Samba配置 674
694 ☆samba-swat 674
695 ☆测试Samba 675
696 ☆smb.conf 675
697 ☆smbusers 687
698 ☆Samba日志文件 687
699 ☆共享访问限制 688
700 ☆通过主机地址限制 688
701 ☆通过用户口令限制 688
702 ☆通过用户名限制 689
703 ☆通过读写限制 690
704 ☆通过浏览限制 690
705 ☆testparm 690
706 ☆testprns 690
707 ☆swat 690
708 ☆nmblookup 690
709 ☆smbstatus 691
710 ☆smbpasswd 691
711 ☆smbclient 691

712 ☆smbprint 694
713 ☆smbtar 694
714 ☆Samba做WINS Server 696
715 ☆Samba做File server 696
716 ☆Samba做Printer Server 697
717 ☆Windows中访问Samba服务器共享 698
718 ☆Samba的Client端 698
719 ☆Samba建立PDC域服务器 699
720 ☆Samba常见故障排除 702
721 ☆samba实验 703
722 ☆NFS 705
723 ☆NFS安装 705
724 ☆NFS启动和停止 706
725 ☆NFS防火墙 706
726 ☆NFS测试 706
727 ☆exports 706
728 ☆etab 708
729 ☆exportfs 708
730 ☆showmount 709
731 ☆NFS客户端 709
732 ☆FTP 710
733 ☆ftp用户管理 711
734 ☆vsFTPd的安装 712
735 ☆vsFTPd启动关闭 712
736 ☆vsFTPd防火墙 712
737 ☆相关配置文件 713
738 ☆vsftpd.conf 714
739 ☆更改FTP PORT 734
740 ☆容许root用户远程登陆 734
741 ☆匿名权限 734
742 ☆本地权限 735
743 ☆更改FTP目录 735
744 ☆实现虚拟路径 736
745 ☆定制欢迎信息 736
746 ☆配置vsFTP限制 737
747 ☆针对不同的使用者的限制 737
748 ☆把用户限制在home中 737
749 ☆绑定IP到vsFTPd 738
750 ☆vsFTPd的日志 738
751 ☆显示vsftp的每一个process 738
752 ☆vsftpd与TCP_wrapper结合 739
753 ☆vsFTPd添加虚拟用户 739

754 ☆FTP错误 740
755 ☆FTP命令 741
756 ☆FTP数字代码的意义 743
757 ☆bc 744
758 ☆xinetd.conf 744
759 ☆tcp-wrapper 745
760 ☆tcpdchk 748
761 ☆tcpdmatch 748
762 ☆SELINUX 749
763 ☆使用身份验证服务前配置 749
764 ☆system-config-authentication 749
765 ☆安装NIS服务器 749
766 ☆配置NIS服务器 749
767 ☆配置NIS辅服务器 750
768 ☆安装NIS客户端 751
769 ☆配置NIS客户端 751
770 ☆NIS客户命令 753
771 ☆LDAP安装 753
772 ☆配置LDAP客户端 753
773 ☆防火墙 754
774 ☆system-config-securitylevel-tui 755
775 ☆iptables入门 756
776 ☆iptables 757
777 ☆iptables启动 758
778 ☆iptables 758
779 ☆iptables表选项 758
780 ☆iptable命令选项 758
781 ☆iptables匹配选项 760
782 ☆iptables动作选项 763
783 ☆iptables中文man文档 764
784 ☆定义默认策略 773
785 ☆查看iptables规则 773
786 ☆增加,插入,删除和替换规则 774
787 ☆清除规则和计数器 774
788 ☆保存恢复iptables规则 774
789 ☆iptables实现NAT 774
790 ☆创建一个简单的防火墙 775
791 ☆iptables应用实例 776
792 ☆NAT禁止访问某些网站 789
793 ☆NAT禁止客户机上网 789
794 ☆NAT禁止客户机访问某些服务 789
795 ☆NAT强制范围某些站点 789

796 ☆NAT禁止ICMP 789
797 ☆NAT发布内部网络服务器 789
798 ☆NAT智能DNS 789
799 ☆PAM 789
800 ☆PAM的文件 790
801 ☆PAM配置 790
802 ☆PAM配置文件的格式 790
803 ☆PAM认证模块 793
804 ☆PAM API 842
805 ☆PAM SPI 843
806 ☆ldd 843
807 ☆system-auth 844
808 ☆access.conf 844
809 ☆限制NIS用户 845
810 ☆定位易被攻击的文件或目录 846
811 ☆tmpwatch 846
812 ☆日志 846
813 ☆utmpdump 848
814 ☆远程日志 849

☆Linux 历史

Linux 出生计划是在 1991 年(是 Linus 计划的),而他真正发布 1.0 版本是在 1994 年，可以说 1994 年，是 Linux 的诞生之日.

1 ☆Linux 正确读音及音标

"Linux"这个单词根据 Linus Torvalds 本人的发音应该是"哩呐克斯",音标是 ['li:n ə ks]重音在"哩"上

2 ☆Linux 相关网站

Redhat 官方站:
http://ww.redhat.com

Fedora 官方站:
http://www.fedora.us/

Mandrakelinux 官方站:
http://www.mandrakelinux.com

SuSE 官方站:
http://www.suse.com

RPM 包查询站:
http://rpmfind.net

kernel.org 镜像站，里面有好多发行版可以下载
http://mirrors.kernel.org

非 RPM 包系统:

Debian 官方站:
http://www.debian.org

Slackware 官方站:
http://www.slackware.com

slackware 软件升级包
http://www.linuxpackages.com/

Gentoo 官方站:

www.gentoo.com

内核官方站:
http://www.kernel.org

3 ☆Linux 安装

win 上面安装的时候
先整理碎片
再 fips
如果是 ntfs 的就运行 ntfsresize

ntfsresize --info /dev/sda1
显示信息

ntfsresize --no-action --size 6000M /dev/sda1
测试会否发生错误

ntfsresize --size 6000M /dev/sda1
更改大小

F2 查看选项,F5 修复模式
linux text
linux noprobe
取消硬件检查,当硬件冲突的时候
linux mediacheck
只检查光盘
linux rescue
修复环境
linux dd
第三方的驱动程序,有一个驱动盘才行
linux askmethod
其他的安装方式,硬盘,网络安装模式
linux updates
升级
memtest86
测试内存

如果设备文件(/dev/...)损坏可重新安装 dev*.rpm

在启动时进入 BIOS 界面

设置系统启动顺序为 A，CDROM，C
修改其他任何推荐的设置
保存并退出 BIOS 设置

使用 Anaconda（图形模式）安装 Red Hat Linux
使用检测到的鼠标配置（除非老师另有指定）
选择全新安装
使用定制安装选项
选择使用 Disk Druid 手动分区，删除所有原有的分区
使用以下分区方案:
/boot 100M
/ 256M
/usr 1000M
swap 512M
/var 400M
格式化所有分区，但是不选择检查坏块
使用默认的启动加载器设置（除非老师另有指定），不创建启动加载器密码
为网络设置选择 DHCP，选择启动时激活
使用默认防火墙配置
选择适当的语言支持
设置对应的时区，根据老师的指示设置 UTC
设置根密码为 redhat
启用 MD5 和 shaow 密码模式（默认验证设置）
选择安装 X window，不选任何其他组件
切换到 tty5 查看文件系统格式化的过程（使用 Ctrl-Alt-5，用 Alt-7 切换回安装向导）
创建启动软盘可选
使用检测到的显示器和图形卡设置（除非老师另有指定）
配置使用图形界面登录
在安装结束后重启，完成初始化设置，不注册 Red Hat Network

安装结束后启动系统，以 root 帐号登录，并检查以下文件:
/var/log/messages
/var/log/dmesg

alt+F1 文本方式的安装界面,有检测信息
alt+F2 可以查看正安装的文件,有简单的 shell,就是 rescue 模式,vi /tmp/X.log
alt+F3 系统消息日志,会显示安装的 log,看系统正在做什么事情,监视安装程序的出错信息,硬件检测信息,就是/tmp/anaconda.log
alt+F4 看内核信息,包括硬件的信息
alt+F5 格式化好前不可用,显示格式化过程
alt+F7 返回图形安装

安装过程中/tmp 下有
anaconda.log 安装的 log,就是 alt+F3
ks.cfg 如果是以 Kickstart 安装的就有
modules.conf 检测到的硬件
netinfo 网络信息
syslog 核心信息,就是 alt+F4
X.log X Window 信息

/mnt/source
是 cd
/mnt/sysimage
是根/

安装好后
/root/install.log
安装日志
/var/log/dmesg
启动信息

11 号分段故障的时候,制定降低内存 linux mem=256M

/RedHat/base/comps.xml
安装包的关系

4 ☆Linux 网络安装

使用 NFS，FTP 或 HTTP 安装 Red Hat Linux

如果您要为 NFS、FTP、或 HTTP 安装设置安装树，您必须复制 RELEASE-NOTES 文件和所有安装光盘的 RedHat 目录下的所有文件。在 Linux 和 UNIX 系统中，以下步骤可以在您的服务器上正确配置目标目录（请对每个光盘重复此步骤）：

插入光盘
mount /media/cdrom
cp -ar /media/cdrom/RedHat <target-directory>
cp /media/cdrom/RELEASE-NOTES* <target-directory>（仅用于第一张光盘）
umount /media/cdrom
这里的 <target-directory> 代表包括安装树的目录路径。

注意
请不要复制附加光盘以及任何其它层次产品光盘的内容。它们会覆盖 Anaconda 正常运行所需的文件。

这些光盘必须在安装 红帽企业 Linux 以后再安装。
.discinfo 这个文件拷贝过去可以保证 system-config-package 可用

破坏现有的系统，重新安装 Red Hat Linux。

破坏现有系统:
cat /var/log/messages > /dev/hda; reboot
cat /var/log/messages > /dev/sda; reboot

重启后使用启动介质启动，按照以下要求安装（由于已经覆盖了分区表，系统将警告没有找到分区表，必须重新初始化）

1. 使用 CD 启动
2. 在 boot 提示下选 linux askmethod
3. 选择对应的语言（English）
4. 在 OK 提示下回车
5. 选择对应的键盘（US）
6. 在 OK 提示下回车
7. 选择对应的安装方式（NFS 镜像，FTP，HTTP）
8. 配置 TCP/IP，选择"使用动态 IP 配置（BOOTP/DHCP）"
9. 在 OK 提示下回车
10. 根据选择的安装方式输入对应的信息:

FTP 方式
FTP 站点名称:192.168.0.254
Red Hat 目录:/inst

HTTP 方
Web 站点名称:192.168.0.254
Red Hat 目录:/inst

NFS 方式
NFS 服务器名:192.168.0.254
NFS 加载点:/mnt/inst

11. 这时 Anaconda 会读取安装镜像并检测显示器和鼠标的类型，显示欢迎界面
12. 选择定制安装
13. 使用 diskdruid 分区。使用以下分区方案:
/boot 100M
/ 2000M
swap 512M
/home 3 × 256M RAID0

14. 启动加载器，时区，图形，防火墙和验证方式都是用默认设置，除非教师指定
15. 设置适当的语言
16. 设置 root 密码为 redhat
17. 安装默认的软件包

5 ☆Linux 启动盘

制作安装盘的工作既可以在 Windows 系统下完成，也可以在 Linux 系统下完成。

在 Windows 系统中，使用 Red Hat 安装光盘中第一张的 rawritewin 程序。此程序在 CDROM 下的 dosutils/rawritewin 目录下。制作第一张软盘时，Image File 请选择 images\bootnet.img ， 这张叫启动盘。制作第二张软盘的时候，Image File 请选择 images\drvnet.net，这张叫驱动盘(Driver Disk)。根据你网卡的型号，驱动盘也许并不需要。

如果是在 Linux 系统中制作安装盘，命令如下:
dd if=/media/cdrom/images/diskboot.img of=/dev/floppy (启动盘)
有可能内核太大,软盘装不下的
[root@panda ~]# dd if=/media/cdrom/images/diskboot.img of=/dev/floppy
dd: 正在写入 ‘/dev/floppy’: 设备上没有空间
读入了 2881+0 个块
输出了 2880+0 个块

cat /mnt/cdrom/images/pcmciadd.img > /dev/fd0
也可以的

mkbootdisk --device /dev/fd0 `uname -r`
也可以的,按当前系统定制,更小

编辑文件 syslinux.cfg
将第一行改写为:
default linux ks=floppy

如果需要用到网卡的驱动盘，则第一行改写为:
default linux ks=floppy dd

此时系统会自动提示插入驱动盘。

6 ☆PXE

PXE(Pre-boot Execution Environment)是由 Intel 设计的协议，它可以使计算机通过网络启动。协议分为 client 和 server 两端，PXE client 在网卡的 ROM 中，当计算机引导时，BIOS 把 PXE client

调入内存执行，并显示出命令菜单，经用户选择后，PXE client 将放置在远端的操作系统通过网络下载到本地运行。

PXE 协议的成功运行需要解决以下两个问题：
既然是通过网络传输，那么计算机在启动时，它的 IP 地址由谁来配置;
通过什么协议下载 Linux 内核和根文件系统

对于第一个问题，可以通过 DHCP Server 解决，由 DHCP server 来给 PXE client 分配一个 IP 地址，DHCP Server 是用来给 DHCP Client 动态分配 IP 地址的协议，不过由于这里是给 PXE Client 分配 IP 地址，所以在配置 DHCP Server 时，需要增加相应的 PXE 特有配置。

至于第二个问题，在 PXE client 所在的 ROM 中，已经存在了 TFTP Client。PXE Client 使用 TFTP Client，通过 TFTP 协议到 TFTP Server 上下载所需的文件。

PXE client 是需要安装 Linux 的计算机，TFTP Server 和 DHCP Server 运行在另外一台 Linux Server 上。Bootstrap 文件、配置文件、Linux 内核以及 Linux 根文件系统都放置在 Linux Server 上 TFTP 服务器的根目录下。

PXE client 在工作过程中，需要三个二进制文件：bootstrap、Linux 内核和 Linux 根文件系统。Bootstrap 文件是可执行程序，它向用户提供简单的控制界面，并根据用户的选择，下载合适的 Linux 内核以及 Linux 根文件系统。

7 ☆PXE 安装

安装 dhcp,ftp 和 tftp 服务器和 system-config-netboot 软件包，并进行适当的设置，使具有 PXE Bootrom 引导功能网卡的电脑（台式机网卡如没有 PXE Bootrom 功能，可买 PXE Bootrom 芯片插入网卡内），在局域网内的所有机器通过网络引导实现三种功能：安装，rescue 和正常运行。

要求：
客户端电脑通过网络引导后进入 boot:
选择界面：install,rescue,local
在客户端输入：install 或直接回车进入安装模式
在客户端输入：linux rescue 进入 rescue 模式
在客户端输入：local 进入正常运行模式

0.安装 dhcp,ftp 和 tftp 服务;

1.设置 vsftpd
/etc/rc.d/init.d/vsftpd 文件用默认值，不用设置。
终端执行以下命令：
#mount /mnt/cdrom
对第一张盘：

#cp /mnt/cdrom/RELEASE-NOTES*.html /var/ftp/pub
#cp -var /mnt/cdrom/RedHat /var/ftp/pub
把另外三张光盘 RedHat/RPMS 目录下的文件都拷至/var/ftp/pub/RedHat/RPMS
service vsftpd start

2.设置 dhcpd
#cp /usr/share/doc/dhcp-3.0.1/dhcpd.conf.sample /etc/dhcpd.conf
在 dhcpd.conf 中加入以下内容：
allow booting;
allow bootp;
class "pxeclients" {
match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
next-server 192.168.152.128;
filename "linux-install/pxelinux.0";
}
就象这样：
ddns-update-style interim;
ignore client-updates;
####################加在这下面
allow booting;
allow bootp;
class "pxeclients" {
match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
next-server 192.168.152.128;
filename "linux-install/pxelinux.0";
}
####################
subnet 192.168.0.0 netmask 255.255.255.0 {

--- default gateway
option routers 192.168.152.128;
option subnet-mask 255.255.255.0;

option nis-domain "domain.org";
option domain-name "domain.org";
option domain-name-servers 192.169.152.129;

option time-offset -18000; # Eastern Standard Time
option ntp-servers 192.169.152.129;
option netbios-name-servers 192.169.152.129;
--- Selects point-to-point node (default is hybrid). Don't change this unless
-- you understand Netbios very well

option netbios-node-type 2;

range dynamic-bootp 192.168.152.12828 192.168.0.254;
default-lease-time 21600;
max-lease-time 43200;

# we want the nameserver to appear at a fixed address 需要为每个无盘站配置静态 ip
host ns {
next-server marvin.redhat.com;
hardware ethernet 12:34:56:78:AB:CD;
fixed-address 207.175.42.254;
}
}

启动 dhcpd 服务：
service dhcpd start

3.设置 tftp
/etc/xinetd.d/tftp 文件用默认值，不用设置。
在终端下输入：
chkconfig --level 345 xinetd on
chkconfig --level 345 tftp on
service xinetd restart
安装 system-config-netboot 软件包，这样会在根目录产生/tftpboot 相关目录和文件：
rpm -ihv system-config-netboot*.rpm

system-config-netboot 可以配置一个 PXE 环境
pxeos -a -i "Enterpirse AS 4" -p NFS -D 0 -s Enterprise4 -L /mnt/inst RHEL4-AS

检查根目录是否生成/tftpboot 子目录下相关目录和文件。
插入第一张光盘
mount /mnt/cdrom
cp /mnt/cdrom/images/pxeboot/*.* /tftpboot/linux-install

4.修改/tftpboot/linux-install/pxelinux.cfg/default 为如下内容：
default install
timeout 100
prompt 1
display msgs/myboot.msg
F1 msgs/myboot.msg
F2 msgs/myboot.msg
F3 msgs/myboot.msg

F4 msgs/myboot.msg
F5 msgs/myboot.msg
F7 msgs/myboot.msg

label install
KERNEL vmlinuz
APPEND initrd=initrd.img

label linux rescue
KERNEL vmlinuz linux rescue
APPEND initrd=initrd.img

label local
localboot 1

5.建立/tftpboot/linux-install/msgs/myboot.msg 文件，内容如下：

^ 09General Boot Help^ 07

You can load the ^ 0cRed Hat^ 07 Enterprise Linux installation program, and
begin the installation process from this prompt. In most cases, the best
way to get started is to simply press the ^ 0f<ENTER>^ 07 key.

##
Type:
install
into install mode.
Type:
linux rescue
into linux rescue mode.
Type:
local
into local linux run mode.
##

^ 05[F1-Main] [F2-General] [F3-Expert] [F4-Kernel] [F5-Rescue]^ 07

至此，服务器设置完成
==========

客户端开引导，直接进入 boot 文本界面
在客户端输入：install 或直接回车进入安装模式
在客户端输入：linux rescue 进入 rescue 模式
在客户端输入：local 进入正常运行模式
到 Installation Method 菜单，选择 ftp 安装
Configure TCP/IP 选择 Use dynamic IP configuration(BOOTP/DHCP)
FTP Setup 中 FTP site name 填入 192.168.152.128
RedHat dir 填入 pub/
就 OK 了

8 ☆system-config-netboot

可以配置无盘环境

9 ☆Kickstart

system-config-kickstart
ksconfig

编辑 /root/anaconda-ks.cfg 文件
保存 anaconda-ks.cfg 文件为 ks.cfg.复制到软盘
用光盘或其他启动介质重启动系统, 把 kickstart 软盘放在软驱中
当出现 boot 提示符时 输入 linux ks=floppy 如果软盘有错系统会提示修正.

10 ☆ks.cfg

ks.cfg 是 Kickstart 安装的核心文件，ks.cfg 放置在软盘的根目中。
所有以"#"号开头的都是注释，可以忽略。

在新安装的 Red Hat 系统下，/root/下有个 anaconda-ks.cfg 文件，可以它为基础进行编辑，也可以用 X-Window 下的 ksconfig 程序进行这项工作。如果读者用 ksconfig 程序进行设定，可能还需要手动进行修改。下面是 ks.cfg 文件的内容

#Generated by Kickstart Configurator
#System language 指明了安装时采用的语言。其实在配置正确的时候，Kickstart 安装不需要人工干预，本文将其定为英语。
lang en_US
#Language modules to install 指明了系统支持的语言环境，如果不安装 X-Window，只需要英语即可。本文增加了对简体中文的支持。
langsupport --default en_US en_US zh_CN.GB231
#System keyboard 指明了系统采用的键盘类型，通常键盘都是 us 兼容行的。
keyboard us

#System mouse 指明了系统采用的鼠标的类型,带滚轮的 PS/2 鼠标应写为:mouse msintellips/2
mouse genericps/2
#System timezone 是时区，中国的用户一般可以选择上海。也可以为 Etc/GMT+8
timezone --utc Asia/Shanghai
#Root password 指明了 root 用户的密码，还有一种形式就是将密码进行加密,--iscrypted 从
/etc/shadow 中拷贝密码
rootpw 12345
#System bootloader configuration
bootloader --location=mbr
#Install Red Hat Linux instead of upgrade 指明了是安装还是升级，如果是升级，应该用 upgrade 代
替 Install 参数。
install
"install"后，手工插入一行“安装文件路径”：
http 方式为：
url --url http://192.168.10.14/data/install
ftp 方式为：
url --url ftp://username:passwd@192.168.10.14/data/install
nfs 方式为：
nfs --server=192.168.10.14 --dir=/data/install
#Use FTP installation media
#指明了安装介质所在的位置，这是 Kickstart 安装的关键之一。安装介质可以放置在
NFS/FTP/HTTP 服务器上，也可以放置在本机硬盘上。
#比如说/tmp/install 下面，如果是用 NFS 安装，则需要把/tmp/install 共享出去，同时要保证将要
安装 Red Hat Linux 的客户机可以访问。只读的权限可以按如下配置:
#/etc/exports 文件如下
#/tmp/install 192.168.10.0/255.255.255.0(ro)
#如果是 FTP 或者 HTTP 安装，则可以把 Red Hat 目录放在 ftproot 或者 httproot 下。值得注意的是，
在 Windows 系统中，可能对"/"和"\"产生歧意，如果 FTP Server 是 Windows 系统，最好将 Red Hat
目录放置到 ftproot 的再下一级目录中。
#参数很多
如
cdrom
harddrive --partition sdb1 --dir /mnt/inst

#Disk partitioning information 描述 Linux 将安装到哪个分区
格式为
part mount dir --size size [--grow] [--maxsize size]
--size 的单位是 MB
选项有: --fstype ext3 --size 4200 --ondisk=sda --grow --maxsize 5000 --noformat

clearpart --all #清除所有分区。
part / --size 4200 #新建一个 4.2GB 的分区作为"/"分区

part swap --size 300 #新建立一个 300MB 的 swap 分区
如果使用已存在的分区，则写为:
partition / --onpart sda1
partition swap -onpart sda2
最好不要用 ksconfig 程序自动设定参数，手工配置比较稳妥。
#Use DHCP networking 还可以--trust=etn0 设置信任的网卡
用 DHCP Server 自动分配 IP 地址。如果装机的数量较多，DHCP 的方式毫无疑问是首选。如果想自己指定 IP 地址，则应该写为:
network --device eth0 --bootproto static --ip 192.168.30.149 --netmask 255.255.255.0 --gateway 192.168.30.113 --nameserver 207.217.120.83,207.217.126.81 --hostname Enterprise3
network --bootproto dhcp
#System authorization information 系统验证用户的方式。本文用的是缺省 Shadow 密码，MD5 方式加密，这是最常见的方式。--enablenis ----nisdimain name -nisserver name --enablekrb5 -enablesmbauth
auth --useshadow --enablemd5
#Firewall configuration 把自定义的防火墙关闭。因为安装 Red Hat 时用的是 ipchains 防火墙，因此笔者建议安装以后采用 iptables 防火墙。 --port=http:tcp --port=443:udp
firewall --disabled
#XWindows configuration information 关于 X-Window 的配置。意思是使用系统自动侦测的配置。Linux 一般都能正确地检测到显示卡和显示器的型号。
#Probe for video card
#Probe for monitor
xconfig --depth 16 --resolution 1024x768 --defaultdesktop=GNOME
这是以 GNOME 做为缺省的窗口管理器，颜色为 16 位色，屏幕分辨为 1024×768。如果在末尾加上"--startxonboot"，就是开机后 X-Window 登陆。
%packages
指明了安装时候选择安装的软件。前面有个"@"符号代表一组 RPM 包，每组包将安装的软件列表，这样软件可以在 Red Hat 第一张安装光盘中的 Redhat/base/comps 文件中找到。如果一个组都不指定，则会安装 Redhat/base/comps 文件中的 Base 组合，也就是最基本的一些 RPM 包，如 Sendmail 等。写一个@Everything 则是完全安装。第 37 行中有单独安装了 lynx 的 RPM 包。
@KDE
@Emacs
lynx

%pre
%pre 表示安装之前执行的命令。
echo " Welcome to my Kickstart"

%post
%post 以后代表安装之后将要执行的命令。
echo "192.168.10.55 Server" >> /etc/hosts

最后的工作是把写好的 ks.cfg 拷贝到软盘上:
cp ks.cfg /mnt/floppy

善用 Kickstart 安装中的%post 命令，就可以直接配置好服务器。
chroot /mnt/sysimage

11 ☆测试 kickstart

mount -t iso9660 -o loop /mnt/cdrom/images/bootdisk.iso /mnt/source
cp -ar /mnt/source/* /tmp/boot
cp ks.cfg /tmp/boot/isolinux
chmod u+w /tmp/boot/isolinux/*
cd /tmp/boot/isolinux
mkisofs -o /tmp/boot1.iso -b isolinux.bin -c boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -R -J -v -T
cdrecorder -v speed=2 dev=0,0,0 /tmp/boot1/iso

或者直从安装光盘启动
linux ks=floppy

linux ks=hd:fd0:/ks.cfg 软盘启动 kickstart
linux ks=cdrom:/ks.cfg 光驱启动 kickstart
linux ks=hd:hda2:/home/mj/ks.cfg 第二分区,第一磁盘上
linux ks=nfs:192.168.17.18:/kicks/ks.cfg
linux ks=http:192.168.17.18:/kicks/ks.cfg

12 ☆Linux 启动顺序

当开始 BIOS 自检
从硬盘开始启动
MBR 就是 446 个字节的一段存储空间,里面存储的是 Bootloader,Linux 的 Bootloader 常见的有 LILO 和 Grub，两者在 MBR 中存储的内容是不同的!
Bootloader 装入
kernel 被装入
读取/etc/inittab #/etc/inittab 是 linux 系统启动时读到的第一个文件，负责确定启动的 level、执行启动脚本 rc.sysinit 和运行进程之父 init.
运行/etc/rc.d/rc.sysinit #/etc/rc.d/rc.sysinit 是 linux 系统启动的时运行的第一个脚本，负责产生系统运行中需要的环境变量和文件系统。其中 mount -a 就是挂接所有在/etc/fstab 中表明的文件系统!
读取/etc/fstab #/etc/fstab 由/etc/rc.d/rc.sysinit 脚本中的 mount -a 调用，出错的机率最大。
运行/etc/rc.d/rc #/etc/inittab 中只确定默认运行的 level，去运行 level 中脚本的其实是/etc/rc.d/rc ，/etc/inittab 中的 level 数字只是/etc/rc.d/rc 的参数

运行/etc/init.d/* #/etc/rc.d/rcX.d 目录下的都是连接，除了 S99local,其他的都是/etc/init.d/*的软连接（符号连接）。
运行/etc/rc.d/rc.local #是系统启动时最后要运行的脚本
启动结束看到 login:提示符

13 ☆MBR

MBR 可通过两种方式运行，其一是定位到活动分区并加载相应的引导扇区，然后将控制权移给此引导扇区，由引导扇区完成该分区内操作系统的基本组件的加载;其二是直接从一指定分区中加载信息，并通过它装入任一分区的操作系统，诸如 LILO、OS/2 boot loader 等引导加载程序都可以配置成这种方式。

14 ☆OS Loader

系统启动引导管理器，是在计算机启动后运行的第一个程序，他是用来负责加载、传输控制到操作系统的内核，一旦把内核挂载，系统引导管理器的任务就算完成退出，系统引导的其它部份，比如系统的初始化及启动过程则完全由内核来控制完成;

启动管理器是存储在磁盘开始扇区中的一段程序，例如，硬盘的 MBR(Master Boot Record)，在系统完成启动测试后，如果系统是从 MBR 启动，则 BIOS(Basic Input/Output System)将控制传送给 MBR。然后存储在 MBR 中的这段程序将运行。这段程序被称为启动管理器。它的任务就是将控制传送给操作系统，完成启动过程.有许多可用的启动管理器，包括 GNU GRUB (Grand Unified Boot Loader),Bootmanager, LILO (LInux LOader), NTLDR (boot loader for Windows NT systems)，等等等.

GRUB 是一个多重操作系统启动管理器--它负责装入内核并引导 Linux 系统。GRUB 还可以引导其它操作系统，如 FreeBSD、NetBSD、OpenBSD、GNU HURD 和 DOS，以及 Windows 95、98、NT 和 2000。

Windows 也有类似的工具 NTLOADER。NTLOADER 就是一个多系统启动引导管理器，NTLOADER 同样也能引导 Linux，只是极为麻烦罢了;

在 Powerpc 架构的机器中，如果安装了 Linux 的 Powerpc 版本，大多是用 yaboot 多重引导管理器，比如 Apple 机目前用的是 IBM Powerpc 处理器，所以在如果想在 Apple 机上，安装 Macos 和 Linux Powerpc 版本，大多是用 yaboot 来引导多个操作系统;

15 ☆GRUB

GNU GRUB 是由 GRUB(GRand Unified Bootloader)派生而来。GRUB 最初由 Erich Stefan Boleyn 设计和应用

GRBU 最早是在 1995 年由 Erich Boleyn 编写的，其最初目的是为了能够从 Utah Mach4 微内核系统(即是现在的 GNU Mach)中启动引导 GNU Hurd 操作系统。现在 GRUB 的基本目的是用于引导符合 Multiboot 标准的操作系统，Multiboot 是一个用于定义符合多引导标准的操作系统规范，这个标准十分全面，但是目前支持它的操作系统仍不多，主要是 GNU Hurd，所以现在仍不能达到 GRUB 所设想的操作系统引导界面大一统的局面，虽然这一天似乎很遥远，但是我们现在仍然可以享受 GRUB 给我们带来的强大功能和高度的灵活性。

GRUB 不但有 Linux 版本，也有 Windows 版本,GRUB 的 Windows 版本是 WINGRUB;

支持大硬盘现在大多数 Linux 发行版本的 lilo 都有同样的一个问题：根分区(/boot 分区)不能分在超过 1024 柱面的地方，一般是在 8.4G 左右的地方，否则 lilo 不能安装，或者安装后不能正确引导系统。而 grub 就不会出现这种情况，只要安装时你的大硬盘是在 LBA 模式下，grub 就可以引导根分区在 8G 以外的操作系统。

LILO 的确是一个选择，现在很多人也都是采用 LILO 解决多引导的问题。但是 LILO 并不易于使用，仍然不够灵活，而且从技术角度而言 LILO 仍然只是一个 linux loader，它并不直接支持其他的操作系统，LILO 对除 LINUX 外其他操作系统所做的只是将其引导块装入，而并不是将这些操作系统的内核装入，这样做的危险之处在于如果某个操作系统分区的引导块不小心被破坏了，那就可能导致这整个系统无法进入，甚至不得不重新安装，GRUB 的推出就是企图改善这种一块硬盘上存在多个引导块的混乱局面。

Grub 可以直接从 FAT、minix、FFS、ext2 或 ReiserFS 分区读取 Linux 内核。这就意味着无论怎样它总能找到内核。另外，GRUB 有一个特殊的交互式控制台方式，可以让您手工装入内核并选择引导分区。这个功能是无价的：假设 GRUB 菜单配置不正确，但仍可以引导系统。

16 ☆GRUB 安装

[root@panda ~]# rpm -ivh grub*.rpm

17 ☆测试 GRUB

[root@localhost ~]# grub
[root@localhost ~]# grub-install

如果您不能找到这两个命令，可能您的可执行程序的路径没有设置;
[root@localhost ~]# /usr/sbin/grub
[root@localhost ~]# /usr/sbin/grub-install

解压、编译和安装 GRUB 源 tar 压缩包时，会将程序 grub 放到 /usr/sbin 中。该程序非常有趣并值得注意，因为它实际上是 GRUB 引导装入器的半功能性版本。是的，尽管 Linux 已经启动并正在运行，您仍可以运行 GRUB 并执行某些任务，而且其界面与使用 GRUB 引导盘或将 GRUB 安装到硬盘 MBR 时看到的界面完全相同。

18 ☆GRUB 引导软盘

cd /boot/grub
dd if=stage1 of=/dev/fd0 bs=512 count=1
dd if=stage2 of=/dev/fd0 bs=512 seek=1

bs 是写入或读出的字节
count 是执行多少次这样的操作
Seek 跳过多少个 bs

或者(推荐使用)
mke2fs /dev/fd0
mount -t ext2 /dev/fd0 /mnt
grub-install --root-directory=/mnt fd0
umount /mnt

或者
mke2fs /dev/fd0
创建了 ext2 文件系统后，需要安装该文件系统：
mount /dev/fd0 /mnt/floppy
现在，需要创建一些目录，并将一些关键文件（原先安装 GRUB 时已安装了这些文件）复制到软盘：
mkdir -p /mnt/floppy/boot/grub
cp /boot/grub/stage1 /mnt/floppy/boot/grub
cp /boot/grub/stage2 /mnt/floppy/boot/grub
现在该使用驻留版本的 GRUB 来设置引导盘的引导扇区了。从 root 用户，输入 "grub"。GRUB 控制台将启动，显示如下：
GRUB version 0.5.96.1 (640K lower / 3072K upper memory)
[Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename.]
grub>
在 grub> 提示符处，输入：
grub> root (fd0)
grub> setup (fd0)
grub> quit
现在，引导盘完成了。在继续下一步骤之前，在看一下刚才输入的命令。
第一个 "root" 命令告诉 GRUB 到哪里查找辅助文件 stage1 和 stage2。缺省情况下，GRUB 会在指定的分区或磁盘上的 /boot/grub 目录中进行查找。在安装引导盘时，也就是几分钟以前，我们已将这些文件复制到正确的位置。

接着，输入了 setup 命令，它告诉 GRUB 将引导装入器安装到软盘的引导记录上;我们将在以后详细讨论这一过程。然后退出。现在，已经制作好引导盘，可以开始使用 GRUB 了。

19 ☆GRUB 硬盘引导

如果要把 GRUB 装到硬盘上，也很容易。这个过程几乎与引导盘安装过程一样。首先，需要决定哪个硬盘分区将成为 root GRUB 分区。在这个分区上，创建 /boot/grub 目录，并将 stage1 和 stage2 文件复制到该目录中，可以通过重新引导系统并使用引导盘，或者使用驻留版本的 GRUB 来执行后一步操作。
我们首先要设置 GRUB 的“根设备”，也就是告诉 GRUB 安装所在的分区：
grub> root (hd0,0)
如果你不能够确定安装 GRUB 所在的分区号的话，可以通过 find 指令查找：
grub> find /boot/grub/stage1
单独分区情况用:
grub> find /grub/stage1
(hd0,0)
GRUB 将会查找文件/boot/grub/stage1 并显示包含这个文件的设备名，当然就是我们安装 GRUB 所在的分区。下面就可以写引导记录了：
grub> setup (hd0)
这条命令将会在第一块硬盘的 MBR 安装 GRUB 引导，如果你不想在 MBR 安装 GRUB，而是希望将 GRUB 安装在分区的引导扇区的话，你可以用下面指令指定安装设备：
grub> setup (hd0,0)
这将会在第一块硬盘的第一个分区的引导扇区安装 GRUB。
OK，现在你就可以重新启动由 GRUB 引导你的系统。

或者
用 install 命令安装 GRUB，明确指定 GRUB 映像的路径。例如：
grub> root (hd0,1)
grub> install /grub/stage1 d (hd0) /grub/stage2 p /grub/menu.lst
install 命令分为三个单独的步骤：
1]将“stage1”安装到 MBR 中。
2]设置“stage2”的地址或者位置。
3]设置一个菜单或选项，用来决定启动哪一个操作系统

现在，假设你将 Linux 安装在了第一块硬盘的第一个分区中或者/dev/hda1 中。记住 GRUB 的命名规则，将上面的名字改为(hd0,0)。键入下面的命令：
grub> install (hd0,0)/boot/grub/stage1 (hd0) (hd0,0)/boot/grub/stage2 p (hd0,0)/boot/grub/menu.conf

install
一个内置的命令，它告诉 GRUB 将(hd0,0)/boot/grub/stage1 安装到 hd0 的主引导纪录中。
(hd0,0)/boot/grub/stage2
告诉 GRUB stage2 映象的位置。

p with the the following options: (hd0,0)/boot/grub/menu.conf
为菜单的显示设置配置文件。

下面是对这个命令用法的总结：
1.install
2.ource_of_stage1
3.where_to_install
4.source_of_stage2
5.p source_of_configuration_file

或者
GRUB 配置过程中的安装 grub-install
grub-install /dev/hd0
grub-install '(hd0)'
grub-install hd0
grub-install /dev/hda
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.
(fd0) /dev/fd0
(hd0) /dev/hda

值得注意的是如果您有一个单独的/boot 分区，应该用如下的办法来安装;
[root@localhost ~]#grub-install --root-directory=/boot /dev/hda

20 ☆GRUB 网络引导

为了使 GRUB 能够支持从网络引导，你需要在编译时打开网络支持选项，关于这个你可以参考源文件目录里的`netboot/README.netboot 为了从网络引导，你首先要在网络设置两个服务，首先是动态 IP 配置服务，可以是 BOOTP，DHCP 或 RARP 服务器，另一个是 TFTP 服务。
然后分别针对不同的服务器 BOOTP，DHCP 或 RARP（三个选一个）运行 bootp,dhcp,rarp。如果一切设置无误的话 GRUB 就会给出 IP，IP netmask 和 TFTP 服务器的 IP 和网关的 IP 地址。最后，从网上得到操作系统的映象文件，网络的设备名称是(nd).如下例：
grub> bootp
Probing... [NE*000]
NE2000 base ...
Address: 192.168.110.23 Netmask: 255.255.255.0
Server: 192.168.110.14 Gateway: 192.168.110.1

grub> root (nd)
grub> kernel /tftproot/gnumach.gz root=sd0s1

grub> module /tftproot/serverboot.gz
grub> boot

21 ☆不同操作系统引导代码

GNU/Hurd

因为 GNU/Hurd 是符合 Multiboot 规范的操作系统，所以非常容易引导:
grub> root (hd0,2)
如果你不记得 Hurd 所在的分区号的话，可以用 find /boot/gnumach 查找。
grub> kernel /boot/gnumach root=hd0s1
grub> module /boot/serverboot
grub> boot

GNU/Linux
---
grub> root (hd1,3)
grub> kernel /vmlinuz root=/dev/hda1
如果你需要指定内核启动参数的话，可以直接加到命令的最后面如：
grub> kernel /vmlinuz root=/dev/hda1 vga=ext
如果你使用 initrd 的话，在 kernel 命令之后执行：
grub> initrd /initrd
grub> boot

FreeBSD

GRUB 能够直接装载.ELF 和 a.out 两种格式的内核，但是由于 FreeBSD 的内核引导接口有时有较大的变动，
所以，对 FreeBSD 最安全的引导方法是引导/boot/loader
grub> root (hd0,a)
grub> kernel /boot/loader
grub> boot

NetBSD,OpenBSD,NetBSD

这三种系统的引导指令序列一样，如下：
1. 'root'设置根设备.
2. `kernel'装载内核.
3. `boot'引导.

DOS/Windows
---

grub> rootnoverify (hd0,0)
grub> chainloader +1
grub> boot

SCO UnixWare
---
grub> rootnoverify (hd1,0)
grub> chainloader --force +1
grub> makeactive //注意这条指令将设置 UnixWare 分区为活动分区，这样要求你的 GRUB 安装在 MBR,否则下次启动时将直接进入 UnixWare 而不会进入 GRUB grub> boot

22 ☆menu.lst

menu.lst 位于/boot/grub 目录中

建立 menu.lst
[root@localhost ~]# touch /boot/grub/menu.lst

再做一个链接/boot/grub/menu.lst 的链接到 /boot/grub/grub.conf
[root@localhost ~]# cd /boot/grub
[root@localhost ~]# ln -s menu.lst grub.conf

menu.lst 的样本:

```
default=0
timeout=5
#splashimage=(hd0,6)/boot/grub/splash.xpm.gz
hiddenmenu
title Fedora Core (2.6.11-1.1369_FC4)
        root (hd0,6)
        kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/ rhgb quite
        initrd /boot/initrd-2.6.11-1.1369_FC4.img
title WinXp
        rootnoverify (hd0,0)
        chainloader +1
```

参数:

default=0
default=0 是默认启动哪个系统，从 0 开始;每个操作系统的启动的定义都从 title 开始的，第一个 title 在 GRUB 的启动菜单上显示为 0,第二个启动为 1，以此类推;

fallback 1

如果默认选项出错，则启动后备选项

timeout=5
等待用户选择菜单项的时间（以秒计），超时则引导默认的选项注:表示在开机后，GRUB 画面出现几秒后开始以默认启动;如果在启动时，移动上下键，则解除这一规则;

splashimage=(hd0,6)/boot/grub/splash.xpm.gz
注:GRUB 的背景画面，这个是可选项;不喜欢 GRUB 的背景画面，所以加#号注掉，也可以删除;支持 640x480,800x600,1024x768 各种模式的开机画面

hiddenmenu
注解:隐藏 GRUB 的启动菜单，这项也是可选的，也可以用#号注掉;

一般的情况下对 Linux 操作系统的启动，一般要包括四行;title 行;root 行;kernel 行;initrd 行;

title XXXXX
注:title 后面加一个空格，title 是小写的，后面可以自己定义,为引导的操作系统的名字

我们过 fdisk -l ;df -lh ; more /etc/fstab 来确认/boot 所在的分区，及 Linux 的根分区所在位置;

查看内核 vmlinuz 的和 initrd 文件名的全称;
[root@localhost ~]# ls -lh /boot/vmlinuz*
-rw-r--r-- 1 root root 1.6M 2005-06-03 /boot/vmlinuz-2.6.11-1.1369_FC4
[root@localhost ~]# ls -lh /boot/initrd*
-rw-r--r-- 1 root root 1.1M 11 月 26 22:30 /boot/initrd-2.6.11-1.1369_FC4.img

root (hd[0-n],y)
通过 root (hd[0-n],y)来指定/boot 所在的分区;
root (hd[0-n],y)表示的是/boot 所在的分区;有时我们安装 Linux 的时候，大多是不设置/boot 的，这时/boot 和/所在的同一个分区;
GRUB 的 root 分区是保存 Linux 内核的分区。
这个 root (hd[0-n],y)很重要，因为/boot 目录中虽然有 grub 目录，最为重要的是还有 kernel 和 initrd 文件，这是 Linux 能启动起来最为重要东西;

root (hd[0-n],y)这一行是可以省略的,root 和 rootnoverify 是一样的作用
把指定/boot 所位于的分区直接写入 kernel 指令行;
这样就省略了通过 root (hd[0-n],y)来指定/boot 所位于的分区;
不省略 root (hd[0-n],y),后面就可以象这样
kernel /vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/
initrd /initrd-2.6.11-1.1369_FC4.img
省略 root (hd[0-n],y),后面就必须
kernel (hd0,5)/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/

initrd (hd0,5)/initrd-2.6.11-1.1369_FC4.img

第一种情况:/boot 和 Linux 的/根分区在同一个分区;
kernel (hd0,6)/boot/vmlinuz-2.6.11-1.1369_FC4 ro root=/dev/hda7
initrd (hd0,6)/boot/initrd-2.6.11-1.1369_FC4.img

第二种情况:/boot 独立一个分区，和 Linux 的根分区不是同一个分区;
kernel (hd0,5)/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/
initrd (hd0,5)/initrd-2.6.11-1.1369_FC4.img

kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/
kernel 一行，是通指定内核及 Linux 的/分区所在位置;
指定 Linux 的内核的文件所处的绝对路径;因为内核是处在/boot 目录中的

如果/boot 是独立的分区，省略 boot,这行要写成:
kernel /vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/

如果/boot 不是独立的一个分区:
kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/

因为/boot 所处的分区已经在 title 下一行 root (hd[0-n],y)中指定了，所以就无需要再指明内核处在哪个分区了

ro
表示只读;

root=LABEL=/
root=Linux 根所位于的分区或标签
来表示 Linux 的/根所处的分区。LABEL=/ 这是硬盘分区格式化为相应文件系统后所加的标签;如果您不了解什么是标签，也可以直接以/dev/hd[a-z]X 或者/dev/sd[a-z]X 来表示;就看您的 Linux 是根分区是在哪个分区了。比如我的是在/dev/hda7 ， 那这里就可以写成 root=/dev/hda7;

如果/boot 是独立一个分区，initrd 一行要把/boot 中省略;如果/boot 不是独处一个分区，而是和 Linux 的/分区处于同一分区，不应该省略;

如果 /boot 不是独处一个分区，而是在/分区一起,应该这样写
initrd /boot/initrd-2.6.11-1.1369_FC4.img

如果 /boot 是单独一个分区,应该这样写
initrd /initrd-2.6.11-1.1369_FC4.img

23 ☆手工 GRUB 引导系统

设定好 kernel 和 initrd 后,输入
grub>boot
就好了

24 ☆GRUB 的 help

```
grub> help
blocklist FILE                                    boot
cat FILE                                           chainloader [--force] FILE
clear                                              color NORMAL [HIGHLIGHT]
configfile FILE                                   device DRIVE DEVICE
displayapm                                         displaymem
find FILENAME                                        geometry DRIVE [CYLINDER HEAD SECTOR [
halt [--no-apm]                                   help [--all] [PATTERN ...]
hide PARTITION                                       initrd FILE [ARG ...]
kernel [--no-mem-option] [--type=TYPE] makeactive
map TO_DRIVE FROM_DRIVE                                md5crypt
module FILE [ARG ...]                              modulenounzip FILE [ARG ...]
pager [FLAG]                                         partnew PART TYPE START LEN
parttype PART TYPE                                 quit
reboot                                             root [DEVICE [HDBIAS]]
rootnoverify [DEVICE [HDBIAS]]                     serial [--unit=UNIT] [--port=PORT] [--
setkey [TO_KEY FROM_KEY]                                setup [--prefix=DIR] [--stage2=STAGE2_
terminal [--dumb] [--no-echo] [--no-ed terminfo [--name=NAME --cursor-address
testvbe MODE                                       unhide PARTITION
uppermem KBYTES                                      vbeprobe [MODE]
```

如果需要得到某个指令的帮助，就在 help 后面空一格，然后输入指令，比如;
grub>help kernel

25 ☆GRUB 命令索引

background 设置图形模式下的背景色。
blocklist 输出文件的块清单标记。
boot 引导已加载的操作系统或扇区链式加载器。
bootp 通过 BOOTP 初试化网络设备。
cat 显示指定文件的内容。
chainloader 加载扇区链式加载器。
clear 清屏幕。

cmp 比较两个文件, 并且报告两者之间的不同的信息。
color 改变菜单的颜色。
configfile 将指定文件作为配置文件予以加载。
debug 打开/关闭除错模式。
default 把 NUM 项菜单设为缺省值。
device 声明 BIOS 驱动器对应的实际物理设备。
dhcp 通过 DHCP 初试化网络设备。
displayapm 显示 APM BIOS 的相关信息。
displaymem 显示 GRUB 所判断到的当前系统的内存分布，包括所有物理内存区域。
dump 显示诸多文件的内容。
embed 如果设备是个驱动器, 则将 Stage 1.5 嵌入到主引导扇区之后。
fallback 如果调用当前的菜单项时出现错误，则转移到 NUM 项后重试
find 在所有分区上查找文件名, 并显示包含该文件的设备。如果设置了参数 --set-root，则在找到第一个匹配后马上停止，并且把该设备设为根。
fontfile 指定中文字体文件，并切换到简体中文显示方式。
foreground 设置图形模式下的前景色。
fstest 切换文件系统的试验模式。
geometry 输出驱动器的相关信息。
gfxmenu 使用 FILE 中的图形菜单。
halt 关闭系统。
help 显示内部命令的帮助信息。
hiddenmenu 隐藏菜单。
hide 通过在分区类型上置隐藏标志，隐藏指定分区。
ifconfig 指定 IP 地址, 子网掩码, 网关和服务器地址。不带参数时，将显示当前的网络配置。
impsprobe 通过一些循环操作，侦测出符合 Intel 多处理器规范 1.1/1.4 的 CPUs，以便于发挥其更好的效能。
initrd 加载 Linux 格式的初始化虚拟盘, 并设置必要的参数。
install 安装 STAGE1 到指定设备上,安装加载 STAGE2 需要的块列表到 STAGE2 上。
ioprobe 侦测指定设备的 I/O 端口号。
kernel 尝试载入主引导影像文件。
lock 如果用户未被认证，则终止命令的执行。
makeactive 将 root 设备置为活动分区。
map 对设备进行映射。
md5crypt 产生一个 MD5 格式的密码。
module 对多重启动影像, 加载启动模块文件 (不处理该文件的内容, 用户必须自己确定核心的要求)。
modulenounzip 与 'module' 类似, 但是自动禁用了解压缩。
pager 没有参数时，切换页模式。
partnew 创建一个新的主分区。
parttype 改变指定分区(PART)的分区类型(TYPE)。
password 设置密码。
pause 终止命令的运行，并给出一段信息。任意键按下后，将继续。

portmap 进行端口映射。
quit 从 GRUB 命令行中退出。
rarp 用 RARP 初始化网络设备。
read 从内存的指定位置读取一个 32-bit 的值，并以十六进制形式显示出来。
reboot 重启系统。
root 设置根分区。
rootnoverify 类似`root'指令, 但不测试安装该分区。这用于有些系统装在 GRUB 能访问的磁盘区之外, 但仍需要设置正确的根分区的情况。有些需要安装分区才能确定的参数可能会有问题。
run 运行一个为光盘启动而设计的引导文件(非模拟方式的)
savedefault 将当前项设置为默认的引导项。
scdrom 寻找系统的第一个 CD-ROM 驱动器，然后为其分配一个驱动器号以备后续使用(--install)。或者，用该驱动器中的可引导光盘启动系统。(--boot)。
serial 初始化一个串口设备。
setkey 改变键盘映射关系。
setup 自动安装 GRUB.
splashimage 图形模式下载入背景图片文件。
terminal 选择一个终端。
terminfo 指定终端的功能。
testload 以多种不同的方式读取文件(由 FILE 指定)的整个内容，并予以比较，以测试文件系统的代码。
testvbe 测试所指定(MODE)的 VBE 模式。
tftpserver 指定 TFTP 服务器的 IP 地址。
timeout 设置在自动启动缺省菜单前所等待的秒数。
title 命名菜单项。
unhide 通过清除隐藏标志，解除指定分区(PARTITION)的隐藏。
uppermem 强制指定仅有(KBYTES) KB 的上位内存。
vbeprobe 侦测 VBE 的信息。
==========

blocklist
用法: blocklist FILE
描述:
输出文件的块清单标记。
显示文件 file 在所占磁盘块的列表。

boot
用法: boot
描述:
引导已加载的操作系统或扇区链式加载器。
仅在命令行模式下需要，当参数都设定完成后，用这条指令启动操作系统

bootp
用法: bootp [--with-configfile]

描述:
通过 BOOTP 初试化网络设备。如果使用了`--with-configfile'参数，此命令将会试图去加载一个特定的配置文件。

cat
用法: cat FILE
描述:
显示指定文件的内容。
显示文件 file 的内容，可以用来得到某个操作系统的根文件系统所在的分区，如下：
grub> cat /etc/fstab

chainloader
用法: chainloader [--force] FILE
描述:
加载扇区链式加载器。若使用了--force 参数, 则忽略该扇区的启动标识的有效性。
把 file 装入内存进行 chainload,除了能够通过文件系统得到文件外，这条指令也可以用磁盘块列表的方式读入磁盘中的数据块,如'+1`指定从当前分区读出第一个扇区进行引导。如果指定了`--force`参数，则无论文件是否有合法的签名都强迫读入，当你在引导 SCO UnixWare 时需要用这个参数。

cmp
用法: cmp FILE1 FILE2
描述:
比较两个文件, 并且报告两者之间的不同的信息。
比较文件的内容，如果文件大小不一致，则输出两个文件的大小，如下：
Differ in size: 0x1234 [foo], 0x4321 [bar]
如果两个文件的大小一致但是在某个位置上的字节不同，则打印出不同的字节和他们的位移：
Differ at the offset 777: 0xbe [foo], 0xef [bar]
如果两个文件完全一致，则什么都不输出。

color
用法: color NORMAL [highlight]
描述:
改变菜单的颜色。Normal 用于指定菜单项的未选中时的颜色，HIGHLIGHT 则用于指定菜单项的被选中时的颜色。如果你未指定 HIGHLIGHT 色，那么我们将使用 NORMAL 的反色值。颜色值的格式是 "FG/BG"。FG 和 BG 是颜色的名称，如下:black(黑), blue(蓝), green(绿), cyan(青), red(红), magenta(粉红), brown(棕), light-gray(亮灰),dark-gray(暗灰), light-blue(浅蓝), light-green(淡绿), light-cyan(淡青), light-red(明红), light-magenta(浅红), yellow(黄) 和 white(白)。注意，BG 的值只能是前八个。另外，若想使用闪烁的前景色，你在 FG 前使用前缀 "blink-" 即可。
改变菜单的颜色，normal 是用于指定菜单中非当前选项的行的颜色，highlight 是用于指定当前菜单选项的颜色。如果不指定 highlight，GRUB 将使用 normal 的反色来作为 highlight 颜色。指定颜色的格式是“前景色/背景色”，前景色和背景色的可选列表如下：
* black

* blue
* green
* cyan
* red
* magenta
* brown
* light-gray
下面的颜色只能用于背景色
* dark-gray
* light-blue
* light-green
* light-cyan
* light-red
* light-magenta
* yellow
* white
你可以在前景色前加上前缀"blink-",产生闪烁效果，你可以在 menu.lst 中加上下面这个选项来改变颜色效果：
title OS-BS like
color magenta/blue black/magenta

configfile
用法: configfile FILE
描述:
将指定文件作为配置文件予以加载。
将 FILE 作为配置文件替代 menu.lst。

debug
用法: debug
描述:
打开/关闭除错模式。

default
用法: default [NUM | `saved']
描述:
把 NUM 项菜单设为缺省值。设置菜单中的默认选项为 num（默认为 0,即第一个选项），超时将启动这个选项

device
用法: device DRIVE DEVICE
描述:
声明 BIOS 驱动器对应的实际物理设备。这条命令只用于 grub 命令行。

在 GRUB 命令行中，把 BIOS 中的一个驱动器 drive 映射到一个文件 file。你可以用这条命令创建一个磁盘映象或者当 GRUB 不能真确地判断驱动器时进行纠正。如下
grub> device (fd0) /floppy-image
grub> device (hd0) /dev/sd0
这条命令只能在命令行方式下使用

portmap
用法: portmap prog_number vers_number
描述:
进行端口映射。

dhcp
用法: dhcp
描述:
通过 DHCP 初试化网络设备。
用 DHCP 协议初始化网络设备。目前而言，这条指令其实就是 bootp 的别名，效果和 bootp 一样。

splashimage
用法: splashimage FILE
描述:
图形模式下载入背景图片文件。

foreground
用法: foreground RRGGBB
描述:
设置图形模式下的前景色。RR 代表红色, GG 代表绿色, BB 代表蓝色。注意他们都使用十六进制的值。

background
用法: background RRGGBB
描述:
设置图形模式下的背景色。RR 代表红色, GG 代表绿色, BB 代表蓝色。注意他们都使用十六进制的值。

clear
用法: clear
描述:
清屏幕。

displayapm
用法: displayapm
描述:

显示 APM BIOS 的相关信息。

displaymem
用法: displaymem
描述:
显示 GRUB 所判断到的当前系统的内存分布，包括所有物理内存区域。
显示出系统所有内存的地址空间分布图。

dump
用法: dump FROM TO
描述:
显示诸多文件的内容。注意，FROM 所指定的必须是一个 GRUB 文件，TO 所指定的必须是一个 OS 文件。

embed
用法: embed STAGE1_5 DEVICE
描述:
如果设备是个驱动器, 则将 Stage 1.5 嵌入到主引导扇区之后。如果是个 FFS 分区, 则可嵌入到该设备饿`引导代码'区中。并输出 Stage 1.5 所占的扇区数。
如果 device 是一个磁盘设备的话，将 Stage1_5 装入紧靠 MBR 的扇区内。如果 device 是一个 FFS 文件系统分区的话，则将 Stage1_5 装入此分区的第一扇区。如果装入成功的话，输出写入的扇区数。

fallback
用法: fallback NUM
描述:
如果调用当前的菜单项时出现错误，则转移到 NUM 项后重试.

find
用法: find [--set-root] FILENAME
描述:
在所有分区上查找文件名, 并显示包含该文件的设备。如果设置了参数 --set-root，则在找到第一个匹配后马上停止，并且把该设备设为根。
在所有的分区中寻找指定的文件 filename，输出所有包含这个文件的分区名。参数 filename 应该给出绝对路径。

fstest
用法: fstest
描述:
切换文件系统的试验模式。
启动文件系统测试模式。打开这个模式后，每当有读设备请求时，输出向底层例程读请求的参数和所有读出的数据。输出格式如下：

先是由高层程序发出的分区内的读请求，输出：<分区内的扇区偏移，偏移(字节数)，长度(字节数)>之后由底层程序发出的扇区读请求，输出：[磁盘绝对扇区偏移] 可以用 install 或者 testload 命令关闭文件系统测试模式。

gfxmenu
用法: gfxmenu FILE
描述:
使用 FILE 中的图形菜单。

geometry
用法: geometry DRIVE [CYLINDER HEAD SECTOR [TOTAL_SECTOR]]
描述:
输出驱动器的相关信息。在 grub 壳程序中, 你可以用这条命令设置驱动器参数为任意值。如果你省略了总扇区数, 则该值缺省有其它参数决定。
输出驱动器 drive 的信息。

halt
用法: halt [--no-apm]
描述:
关闭系统。如果 APM(高级电源管理)功能存在，将使用 APM BIOS 关闭系统，除非指定了 `--no-apm' 选项。

help
用法: help [--all] [PATTERN ...]
描述:
显示内部命令的帮助信息。要查看所有命令的帮助，请使用 `--all' 参数。
在线命令帮助，列出符合 pattern 的命令列表，如果不给出参数，则将显示所有的命令列表。

hiddenmenu
用法: hiddenmenu
描述:
隐藏菜单。

hide
用法: hide PARTITION
描述:
通过在分区类型上置隐藏标志，隐藏指定分区。
这条指令仅仅对 DOS 和 WINDOWS 有用，当在一个硬盘上存在多个 DOS/WIN 的主分区时，有时需要这条指令隐藏其中的一个或几个分区，即在分区表中设置“隐藏”位。

ifconfig
用法: ifconfig [--address=IP] [--gateway=IP] [--mask=MASK] [--server=IP]

描述:
指定 IP 地址, 子网掩码, 网关和服务器地址。不带参数时，将显示当前的网络配置。

impsprobe
用法: impsprobe
描述:
通过一些循环操作，侦测出符合 Intel 多处理器规范 1.1/1.4 的 CPUs，以便于发挥其更好的效能。
检测 Intel 多处理器，启动并配置找到的所有 CPU。

initrd
用法: initrd FILE [ARG ...]
描述:
加载 Linux 格式的初始化虚拟盘, 并设置必要的参数。
为 Linux 格式的启动映象装载初始化的 ramdisk，并且在内存中的 Linux setup area 中设置适当的参数。

install
用法: install [--stage2=STAGE2_FILE] [--force-lba] STAGE1 [d] DEVICE STAGE2 [ADDR] [p] [CONFIG_FILE] [REAL_CONFIG_FILE]
描述:
安装 STAGE1 到指定设备上,安装加载 STAGE2 需要的块列表到 STAGE2 上。如果使用了选项'd', STAGE1 总是试图使用安装 STAGE2 的驱动器, 而不是启动盘。STAGE2 将加载在指定地址上, 如果未声明地址, 则自动检测。如果使用了选项 'p' 或给出了配置文件, 将修改 STAGE2 的第一个数据块, 修正实际 Stage2 启动时使用的配置文件位置。对于 Stage 1.5, 该值为 Stage 2 的路径。如果安装的是 Stage 1.5, 且指定了实际配置文件, 则将该配置文件路径写入 Stage2 中。
这是用来完全安装 GRUB 启动块的命令，一般很少用到。

ioprobe
用法: ioprobe DRIVE
描述:
侦测指定设备的 I/O 端口号。
探测驱动器 drive 所使用的 I/O 口，这条命令将会列出所有 dirve 使用的 I/O 口。

kernel
用法: kernel [--no-mem-option] [--type=TYPE] FILE [ARG ...]
描述:
尝试载入主引导影像文件。其它项将被作为内核的命令行参数而传递给内核。使用此命令以前，内核所用到的模块应该被重新载入。参数 --type 用于说明内核的类型，包括 "netbsd", "freebsd", "openbsd", "linux", "biglinux" 和 "multiboot"。参数 --no-mem-option 用于说明不必自动传递 Linux 的内存参数。
装载内核映象文件(如符合 Multiboot 的 a.out,ELF,Linux zImage 或 bzImage,FreeBSD a.out，NetBSD

a.out 等等）。文件名 file 后可跟内核启动时所需要的参数。如果使用了这条指令所有以前装载的模块都要重新装载。

lock
用法: lock
描述:
如果用户未被认证，则终止命令的执行。

makeactive
用法: makeactive
描述:
将 root 设备置为活动分区。当然，此命令只对 PC 的硬盘主分区有效。
使当前的分区成为活跃分区，这条指令的对象只能是 PC 上的主分区，不能是扩展分区。

map
用法: map [--status] [--hook] [--unhook] [--rehook] [[--read-only] [--fake-write] [--unsafe-boot] [--disable-chs-mode] [--disable-lba-mode] [--heads-per-cylinder=H] [--sectors-per-track=S] TO_DRIVE FROM_DRIVE]
描述:
对设备进行映射。这对于扇区链式引导是很有用的功能，比如 DOS。这里，目的驱动器(TO_DRIVE)可以是一个磁盘文件，即使用磁盘虚拟功能。注意，这要求磁盘文件是连续存放于分区中的。另外，若使用了 --read-only 参数，该功能将使`磁盘'处于只读;若使用了 --fake-write 参数，该功能将使`磁盘'处于假写，即可以"写入"数据，但是却并未记录到真实磁盘上; 若使用了 --unsafe-boot 参数，该功能将使`磁盘'处于真实可写; 若使用了 --disable-chs-mode 参数，CHS 访问功能将被禁用; 若使用了 --disable-lba-mode 参数， LBA 访问功能将被禁用; H 和 S 指定了虚拟磁盘的物理参数。若使用了 --status, --hook, --unhook,--rehook 诸参数之一, 那么其它的命令行参数将被忽略。
映射驱动器 from_drive 到 to_drive。这条指令当你在 chainload 一些操作系统的时候可能是必须的，这些操作系统如果不是在第一个硬盘上可能不能正常启动，所以需要进行映射。如下：
grub> map (hd0) (hd1)
grub> map (hd1) (hd0)

md5crypt
用法: md5crypt
描述:
产生一个 MD5 格式的密码。

module
用法: module FILE [ARG ...]
描述:
对多重启动影像, 加载启动模块文件 (不处理该文件的内容, 用户必须自己确定核心的要求)。剩余参数作为`模快命令行`传递, 象`kernel'命令一样。

对于符合 Multiboot 规范的操作系统可以用这条指令来装载模块文件 file，file 后可以跟这个 module 所需要的参数。注意，必须先装载内核，再装载模块，否则装载的模块无效。

modulenounzip
用法: modulenounzip FILE [ARG ...]
描述:
与 'module' 类似, 但是自动禁用了解压缩。
同 module 命令几乎一样，唯一的区别是不对 module 文件进行自动解压。

pager
用法: pager [FLAG]
描述:
没有参数时，切换页模式。如果使用了 FLAG 参数，那么它为`on' 时为开启, 为 `off' 时为关闭。

partnew
用法: partnew PART TYPE START LEN
描述:
创建一个新的主分区。START 为起始扇区号，LEN 为其包含的扇区数，TYPE 为其分区类型。

parttype
用法: parttype PART TYPE
描述:
改变指定分区(PART)的分区类型(TYPE)。

password
用法: password [--md5] PASSWD [FILE]
描述:
设置密码。当其处于菜单文件的首项时，将禁用所有的交互式菜单编辑功能，包括编辑菜单项(`e`)/进入命令行(`c`)。当正确输入密码 (由 PASSWD 指定)后，载入新的菜单文件(由 FILE 指定)。如果你没有指定 FILE 项，那么上述被禁用的功能将被启用了。当然，你也可以将此命令用到某个菜单项里，用以提高系统安全性。参数 --md5 说明密码(PASSWD)是使用 md5crypt 加密的。
关闭命令行模式和菜单编辑模式，要求输入口令，如果口令输入正确，将使用 new-config-file 作为新的配置文件代替 menu.lst，并继续引导。

pause
用法: pause [MESSAGE ...]
描述:
终止命令的运行，并给出一段信息。任意键按下后，将继续。
输出字符串 message，等待用户按任意键继续。你可以用<^G>(ASCII 码 007)使 PC 喇叭发声提醒用户注意。

quit

用法: quit
描述:
从 GRUB 命令行中退出。
退出 GRUB shell，GRUB shell 类似于启动时的命令行模式，只是它是在用户启动系统后执行 /sbin/grub 才
进入，两者差别不大。

rarp
用法: rarp
描述:
用 RARP 初始化网络设备。

read
用法: read ADDR
描述:
从内存的指定位置读取一个 32-bit 的值，并以十六进制形式显示出来。
从内存的地址 addr 处读出 32 位的值并以十六进制显示出来。

reboot
用法: reboot
描述:
重启系统。

fontfile
用法: fontfile FILE
描述:
指定中文字体文件，并切换到简体中文显示方式。

scdrom
用法: scdrom [--install], [--bootcd]
描述:
寻找系统的第一个 CD-ROM 驱动器，然后为其分配一个驱动器号以备后续使用(--install)。或者，用该驱动器中的可引导光盘启动系统。(--boot)。

run
用法: run FILE
描述:
运行一个为光盘启动而设计的引导文件(非模拟方式的)

root
用法: root [DEVICE [HDBIAS]]
描述:

设置根分区。设置根分区为指定设备(DEVICE), 然后尝试挂接该分区以得到分区大小(用于在 ES:ESI 中传递, 扇区链式启动方式要求这样)。BSD 驱动类型用于启动 BSD 的核心启动), 和确定 BSD 子分区所在的 PC 分区。可选的磁盘偏移参数, 用于 BSD 核心确定有多少个控制器在当前控制器前。比如: 假设同时有一个 IDE 和 SCSI 盘, 而 BSD 根分区在 SCSI 盘上, 那么磁盘偏移就为 1。
将当前根设备设为 device，并且试图 mount 这个根设备得到分区大小。hdbias 参数是用来告诉 BSD 内核在当前分区所在磁盘的前面还有多少个 BIOS 磁盘编号。例如，系统有一个 IDE 硬盘和一个 SCSI 硬盘，而你的 BSD 安装在 IDE 硬盘上，此时，你就需要指定 hdbias 参数为 1。

rootnoverify
用法: rootnoverify [DEVICE [HDBIAS]]
描述:
类似`root'指令, 但不测试安装该分区。这用于有些系统装在 GRUB 能访问的磁盘区之外, 但仍需要设置正确的根分区的情况。有些需要安装分区才能确定的参数可能会有问题。
和 root 类似，但是不 mount 该设备。这个命令用在当 GRUB 不能识别某个硬盘文件系统，但是仍然必须指定根设备。

savedefault
用法: savedefault
描述:
将当前项设置为默认的引导项。

serial
用法: serial [--unit=UNIT] [--port=PORT] [--speed=SPEED] [--word=WORD] [--parity=PARITY] [--stop=STOP] [--device=DEV]
描述:
初始化一个串口设备。UNIT 用于指定要使用的串口设备 (如，0 == COM1); PORT 用于指定端口号; SPEED 用于指定通讯的数率; WORD 为字长; PARITY 为奇偶类型(取 `no', `odd' 和 `even' 之一的值。); STOP 是停止位的长度值; 选项 --device 仅用于命令行模式，用以指定 tty 设备的文件名。默认值是这样的，COM1, 9600, 8N1。

setkey
用法: setkey [TO_KEY FROM_KEY]
描述:
改变键盘映射关系。把 FROM_KEY 映射为 TO_KEY。这里的键必须是字母, 数字, 和以下特殊键: escape(转义), exclam(!), at(@), numbersign(#), dollar($), parenright ()) , caret(^), ampersand(&), asterisk(*), plus(+), percent(%), minus(-), underscore(_), equal(=), parenleft[(], backspace(退格), tab(制表), bracketleft([), braceleft({), bracketright(]), braceright(}), enter(回车), control(控制), semicolon(, colon(, quote('), doublequote("), slash(/), backquote(`), tilde(~), shift(换档), backslash(\), bar(|), comma(,), less(<), period(.), greater(>), question(?), alt(交互), space(空格), capslock(大写), Fx(功能键) 和 delete(删除)。

改变键盘的映射表，将 from_key 映射到 to_key,注意这条指令并不是交换键映射，如果你要交换两个键的映射，需要用两次 setkey 指令,如下：

grub> setkey capslock control
grub> setkey control capslock

其中的键必须是字母，数字或者下面的一些代表某一键的字符串：

`escape', `exclam', `at', `numbersign', `dollar', `percent',
`caret', `ampersand', `asterisk', `parenleft', `parenright',
`minus', `underscore', `equal', `plus', `backspace', `tab',
`bracketleft', `braceleft', `bracketright', `braceright', `enter',
`control', `semicolon', `colon', `quote', `doublequote',
`backquote', `tilde', `shift', `backslash', `bar', `comma',
`less', `period', `greater', `slash', `question', `alt', `space',
`capslock', `FX' (`X' is a digit), and `delete'.

下面给出了它们和键盘上的键的对应关系：

`exclam'=`!'
`at'=`@'
`numbersign'=`#'
`dollar'=`$'
`percent'=`%'
`caret'=`^'
`ampersand'=`&'
`asterisk'=`*'
`parenleft'=`('
`parenright'=`)'
`minus'=`-'
`underscore'=`_'
`equal'=`='
`plus'=`+'
`bracketleft'=`['
`braceleft'=`{'
`bracketright'=`]'
`braceright'=`}'
`semicolon'=`;'
`colon'=`:'
`quote'=`"
`doublequote'=`""
`backquote'=``'
`tilde'=`~'
`backslash'=`\'
`bar'=`|'
`comma'=`,'
`less'=`<'

`period'=`.'
`greater'=`>'
`slash'=`/'
`question'=`?'
`space'=` '

setup
用法: setup [--prefix=DIR] [--stage2=STAGE2_FILE] [--force-lba] INSTALL_DEVICE [IMAGE_DEVICE]
描述:
自动安装 GRUB. 这条命令使用更灵活的 install 命令将 GRUB 安装到指定设备上。如果给出了映象设备,将在该设备寻找 GRUB,否则使用缺省的根设备。根设备可用 root 指令指定。如果你确认系统的 BIOS 应该支持 LBA 模式, 但是 GRUB 却没有工作于该模式, 则请指定 `--force-lba' 参数。如若你在命令行中已安装了一次 GRUB 可是，你却无法卸载 GRUB 程序所在的分区，请指定 `--stage2' 参数。
安装 GRUB 引导在 install_device 上。这条指令实际上调用的是更加灵活但是复杂的 install 指令。如果
image_device 也指定了的话，则将在 image_device 中寻找 GRUB 的文件映象，否则在当前根设备中查找。

terminal
用法: terminal [--dumb] [--no-echo] [--no-edit] [--timeout=SECS] [--lines=LINES] [--silent] [console] [serial] [hercules] [graphics]
描述:
选择一个终端。当指定了多个终端以后, 按任意键方可继续操作。如果控制台和串口都被指定了, 那么你首先在其中按下键盘的终端将被首先选中。如果没有指定任何参数, 那么此命令将显示出当前的终端设置; 参数 --dumb 用以指定一个哑终端, 否则即为 vt100 兼容型; 若使用了 --no-echo 参数, 屏幕上将不会回显输入的字符; 若使用了 --no-edit 参数, the BASH-like 的编辑功能将被禁用; 若使用了 --timeout 参数, 该命令将等待数秒钟(由 SECS 指定); 可使用 --lines 指定最大的行数; 可使用 --silent 选项关闭消息显示。

terminfo
用法: terminfo [--name=NAME --cursor-address=SEQ [--clear-screen=SEQ] [--enter-standout-mode=SEQ] [--exit-standout-mode=SEQ]]
描述:
指定终端的功能。如果此终端为 vt100 兼容型的，则可指定换码顺序 (即使用 \e 代表 ESC, ^X 代表控制码); 在未给任何参数的情况下，将给出当前配置信息。

testload
用法: testload FILE
描述:

以多种不同的方式读取文件(由 FILE 指定)的整个内容，并予以比较，以测试文件系统的代码。输出看起来会有点儿混乱，但是，如果没有错误的话，`i=X, filepos=Y' 里的 X 和 Y 最后必得相等。如果测试即告成功，下一步即可试图载入内核了。
这条指令是用来测试文件系统代码的，它以不同的方式读取文件 file 的内容，并将得到的结果进行比较，如果正确的话，输出的`i=X,filepos=Y`中的 X,Y 的值应该相等，否则就说明有错误。通常这条指令正确执行的话，之后我们就可以正确无误地装载内核。

testvbe
用法: testvbe MODE
描述:
测试所指定(MODE)的 VBE 模式。

tftpserver
用法: tftpserver IPADDR
描述:
指定 TFTP 服务器的 IP 地址。

timeout
用法: timeout SEC
描述:
设置在自动启动缺省菜单前所等待的秒数。

title
用法: title [NAME ...]
描述:
命名菜单项。

unhide
用法: unhide PARTITION
描述:
通过清除隐藏标志，解除指定分区(PARTITION)的隐藏。
仅仅对 DOS/WIN 分区有效，清除分区表中的“隐藏”位。

uppermem
用法: uppermem KBYTES
描述:
强制指定仅有(KBYTES) KB 的上位内存。任何系统的地址变换将被取消。
强迫 GRBU 认为高端内存只有 kbytes 千字节的内存，GRUB 自动探测到的结果将变得无效。这条指令很少使用，可能只在一些古老的机器上才有必要。通常 GRUB 都能够正确地得到系统的内存数量。

vbeprobe

用法: vbeprobe [MODE]
描述:
侦测 VBE 的信息。如果指定了一个模式(MODE 不为空)，则仅显示其信息。

26 ☆GRUB 加口令

明文口令这样设定:
password=123456 /*为 grub 设定修改密码,防止修改 grub 文件
title Fedora Core (2.4.22-1.2061.nptl)
lock #指定锁定的操作系统
#password=654321 /*为系统设置进入密码,单独设置的,有 lock 也可以没有这一行
root (hd0,7)
kernel /boot/vmlinuz-2.4.22-1.2061.nptl ro root=LABEL=/
initrd /boot/initrd-2.4.22-1.2061.nptl.img

MD5 口令这样设定:
用 grub-md5-crypt 成生 GRUB 的 md5 密码
root@linux01 panda]# /sbin/grub-md5-crypt
Password: 在这里输入 123456
Retype password: 再输入一次 123456
$1$7uDL20$eSB.XRPG2A2Fv8AeH34nZ0
在 Grub 中也可以
grub> md5crypt
Password: **********
Encrypted: 1U$JK7xFegdxWH6VuppCUSIb.

然后把 grub.conf 改为
password --md5 $1$7uDL20$eSB.XRPG2A2Fv8AeH34nZ0

口令文件这样设定:
password secret
在这个命令中，“secret”是密码，而/boot/grub/secret-list.conf 是密码文件。这样做之前，你要先进入到根目录或者给出全路径名。
例如：
password secret (hd0,4)/boot/grub/secret-list.conf

想修改就必须按 p 键输入密码

27 ☆GRUB 中的 cat

cat 指令是用来查看文件内容的，有时我们不知道 Linux 的/boot 分区，以及/根分区所在的位置，要查看/etc/fstab 的内容来得知，这时，我们就要用到 cat (hd[0-n],y)/etc/fstab 来获得这些内容;注意要学会用 tab 键命令补齐的功能;

grub> cat (按 tab 键会出来 hd0 或 hd1 之类的;
grub> cat (hd0, 注:输入 hd0,然后再按 tab 键;会出来分区之类的;
grub> cat (hd0,
Possible partitions are:
Partition num: 0, Filesystem type is ext2fs, partition type 0x83
Partition num: 1, Filesystem type unknown, partition type 0x8e
grub> cat (hd0,0)/etc/fstab 注:比如我想查看一下 (hd0,0)/etc/fstab 的内容就这样输入;
LABEL=/ / ext3 defaults 1 1
/dev/devpts /dev/pts devpts gid=5,mode=620 0 0
/dev/shm /dev/shm tmpfs defaults 0 0
/dev/proc /proc proc defaults 0 0
/dev/sys /sys sysfs defaults 0 0
LABEL=SWAP-hda1 swap swap defaults 0 0
/dev/hdc /media/cdrecorder auto pamconsole,exec,noauto,
managed 0 0

可以看到/boot 不是单独分区,和根目录在一起

28 ☆GRUB 引导 Windows

chainloader +1 意思是指定此分区上的第一个磁道来启动
还有 rootnoverify(hd0,0)是让 GRUB 不去 mount hd0,0 分区
据说 win98 一定要用 rootnoverify,不能用 root
Windows 还要用 Makeactive 在 windows 分区上设定 active 标记，就像 chainloader (hdx,y)+1,然后 boot，其中 x,y 用来指明所在分区号。

命令 map:当你有两块硬盘，一个无法从第二块硬盘启动的操作系统，例如 Windowsxp，就可以使用 map 命令.你能够将 hd0 映射为 hd1，将 hd1 映射为 hd0。换句话说，你可以虚拟的交换两个硬盘而启动所需要的操作系统 。

如果你安装了 DOS (或 Windows) 在不是第一块硬盘上,你需要磁盘交换技术(map),因为这些操作系统只能从第一个磁盘启动.
grub> map (hd0) (hd1)
grub> map (hd1) (hd0)

如果您的 Windows 所处于的分区在(hd0,0)，可以在 menu.lst 加如下的一段就能引导起来了;

```
title WinXp
        rootnoverify (hd0,0)
        chainloader +1
```

如果您的机器有两块硬盘，而 Windows 位于第二个硬盘的第一个分区，也就是(hd1,0)

您可以用 grub 的 map 来指令来操作把两块硬盘的序列对调，这样就不用在 BIOS 中设置了;在 menu.lst 中加如下的内容，比如下面的;

```
title WinXp
        map (hd0) (hd1)
        map (hd1) (hd0)
        rootnoverify (hd0,0)
        chainloader +1
makeactive
```

如果 Windows 的分区不位于硬盘的第一个分区怎么办呢？比如在(hd0,2);

这个也好办吧，把 rootnoverify 这行的(hd0,0)改为 (hd0,2)

```
title WinXp
        rootnoverify (hd0,2)
        chainloader +1
makeactive
```

如果 Windows 的在第二个硬盘的某个分区，比如说是位于(hd1,2)，则要用到 map 指令;

```
title WinXp
        map (hd0) (hd1)
        map (hd1) (hd0)
        rootnoverify (hd1,2)
        chainloader +1
makeactive
```

如果有多个 Windows 系统，怎么才能引导出来呢？应该用 hide 和 unhide 指令操作;比如我们安装了两个 Windows ，一个是位于(hd0,0)的 windows 98 ，另一个是安装的是位于(hd0,1)的 WindowsXP;这时我们就要用到 hide 指令了;

```
title Win98
         unhide (hd0,0)
         hide (hd0,1)
        rootnoverify (hd0,0)
        chainloader +1
makeactive
title WinXP
```

```
        unhide (hd0,1)
        hide (hd0,0)
        rootnoverify (hd0,1)
        chainloader +1
makeactive
```

29 ☆从 DOS 引导 linux

找到 ISO 文件里 dosutils 下的 loadlin.exe 和 vmlinuz 两个文件放到同一目录
在 DOS 下进入该目录执行
loadlin vmlinuz root=/dev/hdb6 (注意将 hdb6 更改为你的 linux 启动分区)

用你的 root 身份登录后在命令行下执行
$ grub
grub>root (hd1,5)
grub>setup (hd0) //注意一定要为 setup (hd0) MBR 上
grub>quit
$ reboot

30 ☆GRUB 启动 FreeBSD

```
title FreeBSD 4.0
root (hd0,4,a)
kernel /boot/loader
boot
```

这里我们调用了 FreeBSD 的启动管理器。Root (hd0,4,a)由四个参数，是因为 FreeBSD 对一个单独分区进行了虚拟分割。我们称根分区为“a”。如果 FreeBSD 占据了整个第二块硬盘，这里就应该是 root (hd0,a)。这样，就不是调用内核而是调用 FreeBSD 的启动管理器，它要比调用内核更易使用。（注意：推荐在使用 OpenBSD 和 GNU/Hurd 之前，要先试一下链式加载。）

31 ☆GRUB 命令行模式

进入命令行模式后 GRUB 会给出一个命令提示符`grub>`，此时就可以键入命令，按回车执行。此模式下可执行的命令是在 menu.lst 中可执行的命令的一个子集。此模式下允许类似于 Bash shell 的命令行编辑功能：
<C-f>或<右箭头键> 光标右移一个字符
<C-b>或<左箭头键> 光标左移一个字符
<C-a><HOME> 到这一行的开头
<C-e>或<END> 到行尾
<C-d>或<DEL> 删除光标处的字符
<C-h>或<BackSpace> 删除光标左边的字符

<C-k> 删除光标右边的所有字符（包括光标处的字符）
<C-u> 删除光标左边的所有字符（包括光标处的字符）
<C-y> 恢复上次删除的字符串到光标位置
<C-p>或<向上键> 历史记录中的上一条命令
<C-n>或<向下键> 历史记录中的下一条命令

在命令行模式下<tab>键有补全命令的功能，如果你敲入了命令的前一部分，键入<tab>系统将列出所有可能以你给出的字符串开头的命令。如果你给出了命令，在命令参数的位置按下<tab>键，系统将给出这条命令的可能的参数列表，具体的可用命令集将在后面给出。

32 ☆GRUB 菜单模式

当存在文件/boot/grub/menu.lst 文件时系统启动自动进入此模式。菜单模式下用户只需要用上下箭头来选择他所想启动的系统或者执行某个命令块，菜单的定义在 menu.lst 文件中，你也可以从菜单模式按<c>键进入命令行模式，并且可以按<ESC>键从命令行模式返回菜单模式。菜单模式下按<e>键将进入菜单编辑模式。

菜单编辑模式用来对菜单项进行编辑改变，其界面和菜单模式的界面十分类似，不同的是菜单中显示的是对应某个菜单项的命令列表。如果在编辑模式下按下<ESC>,则取消所有当前对菜单的编辑并回到菜单模式下。在编辑模式下选中一个命令行，就可以对这条指令进行修改，修改完毕后按下<RET>，GRUB 将提示你确认并完成修改。如果你想在当前命令列表中增加一条命令，按<o>在当前命令的下面增加一条指令，按<O>在当前命令前处增加一条指令。按<d>删除一条指令。
b 启动当前行的操作系统
d 删除当前行
e 编辑当前行
o 在当前行下创建空白行
O 在当前行上创建空白行

33 ☆GRUB 菜单隐藏的解除

在开机的时候按一下 Enter 键，就出来了。因为他是被隐藏了;

或者,改配置文件：/etc/grub.conf
找到这行 hiddenmenu ，在这行前面加#号，或者把这行删除;

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
# all kernel and initrd paths are relative to /, eg.
# root (hd0,4)
# kernel /boot/vmlinuz-version ro root=/dev/hda5
```

```
# initrd /boot/initrd-version.img
#boot=/dev/hda
default=0
timeout=5
#splashimage=(hd0,4)/boot/grub/splash.xpm.gz
#hiddenmenu       #就是这一行
title Fedora Core (2.6.11-1.1369_FC4)
        root (hd0,4)
        kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/ rhgb quiet
        initrd /boot/initrd-2.6.11-1.1369_FC4.img
title WinXp
        rootnoverify (hd0,0)
        chainloader +1
```

34 ☆GRUB 丢失或损坏的策略

```
grub>cat (hd0,0) /root/grub/grub.conf(为了看参数。)
grub>root (hd0,1)
grub>kernel (hd0,0) /boot/vmlinuz-2.4.18-11 ro root=/dev/hda2
grub>initrd (hd0,0) /boot/initrd-2.4.18-11.img
grub>boot
OK!
```

1、运行 GRUB，重建 MBR（见其它的贴子）。
2、在开机就进入 grub>的情况下，grub>root (hdx,y) ;grub>setup hd(0)也可重建 MBR。

进去 Linux 的 rescue 模式！
用软盘或光盘启动，然后在启动的提示符输入:linux rescue
按照提示进入一个 Shell 状态，你可以到/mnt/下面看到一个 sysimage 这么目录，进去以后，就是你安装 linux 的/分区.
使用命令将根分区变为当前目录的根分区:chroot /mnt/sysimage
然后转到/sbin/这个目录中.
使用 fdisk -l 显示当前分区情况，然后使用#grub-install /dev/hdx(x 为你使用的是那块硬盘安装的，一般情况下是 hda)
使用 exit 推出 chroot，再使用 exit 退出 linux rescue 模式，系统将重新启动！取出光盘，应该可以看到 grub 安装好了.
在具体的环境中，编辑/boot/grub/grub.conf 文件和 menu.lst 文件.

1、由于重新安装 Windows 或其它未知原因而导致 GRUB 的丢失;

您可以通过系统安装盘、livecd 进入修复模式;

首先:您根据前面所说 grub-install 来安装 GRUB 到/boot 所在的分区;要仔细看文档，/boot 是不是处于一个独立的分区是重要的，执行的命令也不同;

其次:要执行 grub ，然后通过 root (hd[0-n],y)来指定/boot 所位于的分区，然后接着执行 setup (hd0)，这样就写入 MBR 了，比如下面的例子;

grub>root (hd0,6)
grub>setup (hd0)
grub>quit

重新引导就会再次出现 MBR 的菜单了或命令行的提示符了;

2、如果出现 GRUB 提示符，而不出现 GRUB 的菜单，如何引导系统;
存在的问题可能是/boot/grub/menu.lst 丢失，要自己写一个才行;您可以用命令行来启动系统，进入系统后写一写 menu.lst 就 OK 了。前面已经谈过了;

写好后还要建一个 grub.conf 的链接，如下:
[root@localhost ~]# cd /boot/grub
[root@localhost grub]# ln -s menu.lst grub.conf

35 ☆linux 的启动

无论如何，以下是系统启动必经的过程：
1，启动器（如 grub）在硬盘上找到内核的镜像，把它导入内存中启动;
2，由内核初始化设备以及启动它们的驱动程序;
3，内核加载根文件系统（root filesystem）;
4，内核启动初始化程序 initrd;
5，由 initrd 设置一系列在配置文件中配置的进程;
6，最后 initrd 按顺序启动这些进程直至给出登录画面。

36 ☆initrd

initrd -- 由启动加载器进行初始化的 RAM DISK

init 没有什么特殊的，它也是一个跟其他应用程序差不多的程序，在/sbin 里也能够找的到。
init 的主要作用其实就是是有次序地启动或者关闭一些进程。
主要是为了解决 vmlinuz 太大的问题，用 initrd 可以解决这个问题。

initrd 的进程号为 1,是所有进程的父进程，内核初始化完毕之后，initrd 程序开始运行。其他软件也同时开始运行。initrd 程序通过/etc/inittab 文件进行配置

/dev/initrd 这个特殊文件是一个只读的块设备文件。/dev/initrd 设备文件是一个在内核被启动之前由启动加载器进行初始化的 RAM disk。

随后，内核利用/dev/initrd 设备文件的内容进行两个阶段的(系统)自举。

在(系统)自举的第一个阶段，内核进行初始化，根据/dev/initrd 的内容挂载一个原始根文件系统。在第二个阶段，一些附加的驱动或者其他模块从原始的根设备中被加载。在加载完附加模块后，一个新的根文件系统(也就是常规的根文件系统)从别的设备被挂载。

自举操作流程(参见 man initrd)
使用 initrd 进行系统自举，系统初始化如下：
1.启动加载器把内核程序以及/dev/initrd 的内容加载到内存
2.在内核初始化过程中，内核把/dev/initrd 设备的内容解压缩并拷贝到/dev/ram0 设备上，随之释放被/dev/initrd 占用的内存空间
3.接着内核以可读写的方式把/dev/ram0 设备挂载为原始的根文件系统
4.如果(不知道如何翻译 indicated)常规根文件系统也是原始根文件系统(举例来说，/dev/ram0)，那么内核跳至最后一步正常启动
5.如果可执行文件/linuxrc 存在于原始根文件系统上，/linuxrc 就以 uid 为 0 的帐户身份被执行。(/linuxrc 文件必须具有可执行属性，它可以是包括 shell 脚本在内的任何有效的可执行文件)
6.如果/linuxrc 没有被执行或者当/linuxrc(的运行)终止时，常规根文件系统被挂载。(如果/linuxrc 退出时在原始根文件系统上挂载了任意文件系统，那么内核的行为则是不定的。阅读注意事项以确定当前的内核行为)
7.如果常规根文件系统存在/initrd 目录，那么/dev/ram0 将从/移动到/initrd。否则如果/initrd 目录不存在，/dev/ram0 将被卸载。(当从/移动到/initrd 而/dev/ram0 没有被卸载时，会导致进程仍能从/dev/ram0 运行)。如果/initrd 目录不存在，并且当/linuxrc 退出时任何进程仍能从/dev/ram0 运行，内核的行为是不定的。阅读注意事项以确定当前内核的行为。)
8.正常的启动过程(比如/sbin/init 的调用)将在常规根文件系统上进行

注意事项
1.在当前内核下，当/dev/ram0 从/被移动到/initrd 时，任何已挂载的文件系统依然能被访问。然而，/proc/mounts 条目不会被更新。
2.在当前内核下，如果/initrd 不存在，如果/dev/ram0 被其他进程使用中或者有任何文件系统被挂载其上，/dev/ram0 将不会被完全卸载。如果/dev/ram0 没有被完全卸载，那么/dev/ram0 将驻留在内存

1,如果没有/dev/initrd 设备，用如下命令创建：
#mknod -m 400 /dev/initrd b 1 250
#chown root:disk /dev/initrd
2,要使用 initrd，编译内核时必须选择以下两项：
CON-FIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_INITRD=y
当使用/dev/initrd 时， RAM disk driver 不能作为模块加载。

37 ☆inittab

代码:
#
inittab This file describes how the INIT process should set up the system in a certain run-level.
#
Author: Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org> Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
runlevel 用来表示在 init 进程结束之后的系统状态，在系统的硬件中没有固定的信息来表示 runlevel，它纯粹是一种软件结构。init 和 inittab 是 runlevel 影响系统状态的唯一原因。机器的运行状态可以用运行级别来描述，运行级别(Runlevels)有七种，用 0 到 6 表示。系统通常只运行在一个 runlevel 下面。在关机的时候，init 会把系统终端引导入另一个运行级别下，告诉 kernel 进行关机操作，关闭系统服务等等。在上述例子中 inittab 文件起始阶段的注释主要用来描述 runlevel：

Runlevel 0 是让 init 关闭所有进程并终止系统。
Runlevel 1 是用来将系统转到单用户模式，单用户模式只能有系统管理员进入，在该模式下处理那些在有登录用户的情况下不能进行更改的文件，改 runlevel 的编号 1 也可以用 S 代替。
Runlevel 2 是允许系统进入多用户的模式，但并不支持文件共享，这种模式很少应用。
Runlevel 3 是最常用的运行模式，主要用来提供真正的多用户模式，也是多数服务器的缺省模式。
Runlevel 4 一般不被系统使用，用户可以设计自己的系统状态并将其应用到 runlevel 4 阶段，尽管很少使用，但使用该系统可以实现一些特定的登录请求。
Runlevel 5 是将系统初始化为专用的 X Window 终端。对功能强大的 Linux 系统来说，这并不是好的选择，但用户如果需要这样，也可以通过在 runlevel 启动来实现该方案。
Runlevel 6 是关闭所有运行的进程并重新启动系统。

在 inittab 文件中以#开头的所有行都是注释行。注释行有助于用户理解 inittab 文件，inittab 文件中的值都是如下格式：
label:runlevel:action:process

label
label 是 1~4 个字符的标签，用来标示输入的值。一些系统只支持 2 个字符的标签。鉴于此原因，多数人都将标签字符的个数限制在 2 个以内。该标签可以是任意字符构成的字符串，但实际上，某些特定的标签是常用的，在 Red Hat Linux 中使用的标签是：
代码:
id 用来定义缺省的 init 运行的级别
si 是系统初始化的进程
ln 其中的 n 从 1~6,指明该进程可以使用的 runlevel 的级别
ud 是升级进程
ca 指明当按下 Ctrl+Alt+Del 是运行的进程
pf 指当 UPS 表明断电时运行的进程

pr 是在系统真正关闭之前，UPS 发出电源恢复的信号时需要运行的进程
x 是将系统转入 X 终端时需要运行的进程

runlevel
runlevel 字段指定 runlevel 的级别。可以指定多个 runlevel 级别，也可以不为 runlevel 字段指定特定的值。

action
action 字段定义了该进程应该运行在何种状态下：
代码:
boot 在系统启动时运行，忽略 runlevel
bootwait 在系统启动时运行，init 等待进程完成。忽略 runlevel
ctrlaltdel 当 Ctrl+Alt+Del 三个键同时按下时运行，把 SIGINT 信号发送给 init。忽略 runlevel
initdefault 不要执行这个进程，它用于设置默认 runlevel
kbrequest 当 init 从键盘中收到信号时运行。这里要求键盘组合符合 KeyBoardSigral(参见 /usr/share/doc/kbd-*关于键盘组合的文档)
off 禁止进入，因此该进程不运行
once 每一个 runlevel 级别运行一次
ondemand 当系统指定特定的运行级别 A、B、C 时运行
powerfail 当 init 收到 SIGPWR 信号时运行
powerokwait 当收到 SIGPWD 信号且/etc/文件中的电源状态包含 OK 时运行
powerwait 当收到 SIGPWD 信号，并且 init 等待进程结束时运行
respawn 不管何时终止都重新启动进程
sysinit 在运行 boot 或 bootwait 进程之前运行
wait 运行进程等待输入运行模式

process
process 字段包含 init 执行的进程，该进程采用的格式与在命令行下运行该进程的格式一样，因此 process 字段都以该进程的名字开头，紧跟着是运行时，紧跟着是运行时要传递给该进程的参数。比如/sbin/shutdown -t3 -r now，该进程在按下 Ctrl+Alt+Del 时执行，在命令行下也可以直接输入来重新启动系统。

特殊目的的记录

仔细学习例子文件，学习应用其中关于 inittab 的语法格式。该文件的大多数内容都可以忽略，因为超过一半的内容都是注释，剩余的一些文件内容主要是用来实现某些特殊的功能：

id 的值表明缺省的 runlevel 是 3。
ud 的值可以唤醒/sbin/update 进程，该进程为保持磁盘的完整性，将在对磁盘进行 I/O 操作之前清空整个 I/O 缓冲区。
pf、pr 和 ca 的值只被特定的中断所调用。
如果系统是专用的 X 终端，则只需 x 的输入值。

getty 进程来提供虚拟终端设备的服务，例如：
3:2345:respawn:/sbin/mingetty tty3
标签字段的值是 3,3 是设备 tty3 的数字后缀,tty3 与相应的进程相关联，该 getty 进程可以启动的 runlevel 是 2、3、4 和 5,当该进程终止时，init 马上就重新启动它。启动进程的路径名是/sbin/mingetty，该进程是实现虚拟终端支持的最小版本的 getty，为 tty3 提供启动虚拟设备的进程。

si::sysinit:/etc/rc.d/rc.sysinit
该值告诉 init 程序运行/etc/rc.d/rc.sysinit 脚本文件来初始化系统，该脚本文件与所有启动的脚本类似，它只是一个包含 Linux 的 shell 命令的可执行文件，注意输入的字符串必须包括该脚本的完整路径。不同版本的 Linux 存放该脚本的位置也不相同，但不用刻意去记忆这些位置，只需查看/etc/inittab 文件即可，该文件中包含启动脚本文件的确切位置。

l3:3:wait:/etc/rc.d/rc 3
该行表示要启动所有由 runlevel 3 定义的支持多用户的进程的服务，标签 l3 是级别 3 的标志，在与运行 level3 相关的 inittab 文件之前 init 程序将一直处于等待状态，直到启动脚本终止，init 将执行/etc/rc.d/rc，并向该脚本传递命令行的参数为 3。

Default runlevel. The runlevels used by RHS are:
0 - halt (Do NOT set initdefault to this)
1 - Single user mode
2 - Multiuser, without NFS (The same as 3, if you do not have networking)
3 - Full multiuser mode
4 - unused
5 - X11
6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
sysinit 是 init 首先需要执行的动作，在选择运行级别之前就已经运行了。

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

这几行非常重要，因为它是用来触发大部分的系统服务和配置的。这些系统服务以及配置信息都保存在 rc*.d 和 init.d 目录中。上面的例子里 init 设置为在运行级别 5 下运行/etc/rc.d/rc 5 命令，采取的动作(action)是 wait。整个一行的意思就是：如果进入运行级别 5 则运行 rc 5 命令，并且将一直等待(wait)到这个命令运行完毕后再做其他。
rc 的意思就是 run commands
在 runlevel 5 中，命令一般会在/etc/rc.d/rc5.d 或者/etc/rc5.d 这两个目录下。同理，runlevel 1 的命令在 rc1.d 下，runlevel 2 的在 rc2.d 下，依此类推。在 rc5.d 目录下你能看到这些东西：
K01yum K35smb K74nscd S06cpuspeed
K02cups-config-daemon K35vncserver K74ntpd S08iptables
K02NetworkManager K35winbind K75netfs S10network
K02NetworkManagerDispatcher K50netdump K85mdmonitor S12syslog
K03rhnsd K50snmpd K85mdmpd S18auditd
K05saslauthd K50snmptrapd K86nfslock S26lm_sensors
K10cups K50vsftpd K87portmap S44acpid
K10psacct K66mDNSResponder K89named S50hsf
K15gpm K67nifd K89netplugd S90crond
K15httpd K68rpcidmapd K89rdisc S90xfs
K17iiim K69rpcgssd K90bluetooth S95anacron
K20nfs K69rpcsvcgssd K91isdn S95atd
K24irda K72autofs K94diskdump S97messagebus
K25sshd K73ypbind K96pcmcia S98haldaemon
K30sendmail K74apmd S05kudzu S99local

在这个运行级别的目录下的命令运行的时候，都是采用下面的模式：
s10sysklogd start
s12kerneld start
s15netstd_init start
s18netbase start
.....
注意每一个运行命令，命令中的 S 表示这个命令是在系统启动的时候运行，而数字(00 到 99)表示这个命令在所有 rc 中的顺序。这些 rc*.d 命令一般都是 shell 脚本，用来运行/sbin 和/usr/sbin 中的程序。一般来说，通过观察脚本命令的组成就可以知道这个脚本到底运行了哪个程序。
S 和 K 的启动序号必须唯一的.
同样，我们也可以手工启动一些服务。比如，如果你想手工启动 httpd 服务程序，运行 s99httpd start 就可以了。而且，你也可以非常简单地关闭某个服务，直接对 rc*.d 目录下的命令使用 stop 参数：如 s99httpd stop 即可。
有一些 rc*.d 目录里有部分 K 开头的命令，意思就是 kill。这样的话，rc 运行命令就是使用的 stop 参数而不是 S 所表示的 start。这种情况一般出现在系统关机的时候，还是比较容易理解的。

添加删除服务
如果你想要添加删除或者配置 rc*.d 目录下的服务程序，那么就得先细看一下这个目录里的文件了，使用 ls -l 命令将可以看到：

lrwxrwxrwx 1 root root 15 Aug 27 18:18 K03rhnsd -> ../init.d/rhnsd
lrwxrwxrwx 1 root root 15 Aug 27 09:39 K15httpd -> ../init.d/httpd
lrwxrwxrwx 1 root root 14 Oct 11 20:39 K17iiim -> ../init.d/iiim
lrwxrwxrwx 1 root root 13 Aug 27 09:39 K20nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 14 Aug 27 09:39 K24irda -> ../init.d/irda
lrwxrwxrwx 1 root root 14 Oct 11 20:39 K25sshd -> ../init.d/sshd
lrwxrwxrwx 1 root root 18 Aug 27 11:53 K30sendmail -> ../init.d/sendmail
lrwxrwxrwx 1 root root 13 Aug 27 09:40 K35smb -> ../init.d/smb
.....

这个目录下的命令实质上就是一些链接，几乎都是指向 init.d 目录。从目录前面的 ..可以推测，一般都是在/etc/init.d 或者/etc/rc.d/init.d 下。linux 发行版包含这些链接的目的其实就是为了使各个运行级别(runlevel)下启动脚本统一起来，除了可以使系统的组织布局更简单一些没有什么特别的意义。
从当前的运行级别中去除某个 init.d 脚本命令而不让它运行，最容易想到的办法就是在相应的 rc*.d 目录里删除链接(symbolic link)，不可否认这的确可行，不过如果出了什么差错的话那可就惨了，毕竟这些名字都不短，出了差错想重新添加链接名字很难记清楚。所以，不要直接从 rc*.d 目录下删除链接，而是在前面加上一个下划线标记就可以了：mvs99httpd_s99httpd。这样的话，在系统启动的时候，rc 就会忽略_s99httpd 因为它不是 S 或 K 开头的，而且这样出了问题还可以非常直接的修复。
要添加一个服务，那么就需要在 init.d 目录里自己创建一个脚本，然后在 rc*.d 目录里创建一个指向它的链接。不过，最简单的方法莫过于参照 init.d 目录里已经有的脚本，然后予以修改即可。添加服务的时候，需要注意的是应该给你的服务设定一个合适的启动顺序。如果你自己添加的服务启动太早的话可能会无效，因为服务之间存在着一定的依赖关系。对于一些非基本的服务，大部分系统管理员会比较倾向于使用 90 以后的顺序，那个时候大部分系统服务已经启动完毕了。linux 的发行版本通常都自带了一个添加删除系统服务的命令。比如 debian 版本里的 update-rc.d 命令;redhat 里的 chkconfig 命令。相应的 GUI 程序可以让服务配置更加直观，利于管理。

```
# Things to run in every runlevel
ud::once:/sbin/update

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

ctrlaltdel 控制的是在系统文本终端下按下 Ctrl+Alt+Del 键所执行的操作，在大部分系统里这个被设置为重新启动。

```
# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
```

If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
respawn 动作可以让 init 运行后面的命令，并且命令一旦停止运行 respawn 会再次执行这个命令，也就是说循环运行后面的命令。getty 程序是用来给出登录提示符的，上面一行指示的是第一虚拟控制台，即按下 ALT+F1 或者 CONTROL-ALT-F1 所进入的终端界面。respawn 可以使终端提示符在登录以后一直显示出来。

Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon

38 ☆init

init 1-5
可以切换运行级别

控制 init
有时候，我们可以稍稍设置一下 init 改变运行级别(runlevel)，重新审视一下 inittab 文件或者关闭一次系统。由于 init 是整个系统中第一个运行的进程，所以它的 ID 号永远都为 1。控制 init 可以用 telinit 命令，假设要转换到 runlevel 3 下：telinit 3
运行级别转换的时候，init 会关闭所有那些没有被新的 runlevel 所定义的进程，因此改变运行级别的时候要加以小心。当你需要添加或者去除 inittab 文件里的一些命令(如 respawn 类的)时，需要通知 init 这些改变以使它重新读取一次 inittab 配置文件。一些人通过 kill -HUP 1 来实现，这个方法适用于大部分 unix 版本的系统上，不过也可以用下面面的 telinit 命令：
telinit q
重新执行这个文件/etc/inittab

init 1 和 init S(telinit 1 和 telinit S)是不一样的，init S 只是简单让当前终端运行到单用户模式下，关闭多用户模式下的其他终端，并没有到/etc/rc1.d/中执行任何脚本，各种多用户下的服务仍然在运行，而运行 init 1 时才真正到/etc/rc1.d 下去执行，而这里的 1 是受 /etc/inittab 中控制的，按照正常配置，当运行 init 1 时，起动脚本/etc/rc 1 并且到/etc/rc1.d/下执行，当运行到 S00single 后，似乎是 init 自己调用了 init S 把当前终端引入单用户模式下，原因在于/etc/rc1.d/S00single 中最后一句

exec init -t1 S，在等级 1 模式下为什么没有看到 login 或者说为什么没有起动 getty,其实很简单，在/etc/inittab 中"Run getttys in standard runlevels"中，并没有要求等级 1 运行一个 getty 像这样 1:2345:respawn:/sbin/mingetty tty1 只要在“2345”中加上“1”。便可以起动个 miigetty,但又当/etc/rc1.d/S00single 脚本中最后会运行 exec init -t1 S，所以这些已经起动好了的 getty 又会被关闭。大致就是这种情况，我觉得 init S 或者 telinit S 所做的最重要的动作应该就是关闭由/etc/inittab 中起动的 getty.还有一点，当进入单用户模式后如果输入 exit 退出，则 init 会根据/etc/inittab 中设置的 initdefault 字段进入默认运行等级，如果没有，init 会提示输入一个运行等级

39 ☆chkconfig

管理 linux 启动服务

chkconfig --help
使用--help 查看 chkconfig 语法信息

chkconfig --list [name]
使用 chkconfig 检查系统服务的状态

chkconfig --add name
增加服务

chkconfig --del name
删除服务

chkconfig [--level levels] name <on|off|reset>
打开关闭服务

chkconfig [--level levels] name

观察 on 和 --add 的差异, off 和 --del 的差异
chkconfig isdn --list
chkconfig isdn on
chkconfig isdn --list
chkconfig isdn off
chkconfig isdn --list
chkconfig isdn --del
chkconfig isdn --list
chkconfig isdn --add
chkconfig isdn --list

40 ☆检查运行服务

chkconfig --list | grep on

netstat -tualp
(TCP, UDP, ALL, Listen, PID)

netstat -taupe
(TCP, UPD, ALL, PID, Extended INFO)

netstat -tulpe
查看你的哪些进程正在监听着哪些端口

nmap -sSUR -P0 -vO stationx
(portmap, UDP, TCP stealth scan)

ps -auxw or top
查看系统执行序的方式

nmap -sSUR -P0 -vO <stationX>
扫描端口

41 ☆系统服务详解

acpid
配置文件：/proc/acpi/event
预设端口：无
说明：Advanced Configuration and Power Interface，为替代传统的 APM 电源管理标准而推出的新型电源管理标准。
是否需要启动：如果你需要对电源进行管理，那就需要启动。

alsasound
Alsa 声卡驱动程序支持。Alsa 声卡驱动程序本来是为了 一种声卡 Gravis UltraSound(GUS)而写的，该程序被证明很优秀，于是作者就开始为一般的声卡写驱动程序。Alsa 和 OSS/Free 及 OSS/Linux 兼容，但是有自己的接 口，甚至比 OSS 优秀。

amd
运行 automount 守护程序，该守护在必要时自动安装一些 本地设备和 NFS 文件系统。

anacron
不考虑系统 downtime 期间的 cron 服务
配置文件：/etc/anacron

预设端口：无
说明：一个自动化运行任务。Red Hat Linux 随带四个自动化任务的工具：cron、 anacron、at、和 batch。当你的 Linux 主机并不是全天候开机，这个 anacron 就可以帮你执行在"crontab"设定的时间内没有执行的工作。举例来说，当你的主机在晚上 12:00 会自动关闭，但是偏偏 crontab 这个例行性工作是在 4:00 工作，这个时候例行性工作就不能起作用了。不过利用 anacron 就能做到。
是否需要启动：如果主机已经 24 小时开机，而且运行了 cron，那么这个守护程序就不需要启动了。

apmd
apmd 用来监视系统用电状态，并将相关信息通过 syslogd 写入日志。也可以用来在电源不足时关机。
配置文件：/etc/sysconfig/apmd
预设端口：无
说明：Advanced Power Management，高级电源管理。传统的电源管理标准。一般系统都会同时支持 APM 和 APMD 两种标志，但系统加载时只需加载一个即可。对于笔记本电脑比较有用，可以了解系统的"电池电量"。
是否需要启动：如果我们使用的是台式电脑或一直开机的机型，就不需要使用这个守护程序。

arpwatch
该程序主要用来维护以太网物理地址和 IP 地址的对应关系。

atalk
AppleTalk 守护程序。注意不要在后台运行该程序，该程序的数据结构必须在运行其他进程前先花一定时间初始化。

atd
运行用户用 At 命令调度的任务。也在系统负荷比较低时运行批处理任务。用于 at 和 batch 的服务
配置文件：/etc/at.allow，/etc/at.deny
预设端口：无
说明：一个自动化运行任务。
是否需要启动：通常需要启动。不过如果你一直使用 cron，那么也可以不启动。

autofs
自动安装管理进程 automount，与 NFS 相关，依赖于 NIS
配置文件：/etc/rc.d/init.d/autofs
预设端口：无
说明：实现光盘、软盘的自动加载。
是否需要启动：一般不需要启动。

bootparamd
该服务允许老的 Sun 工作站从 Linux 网络启动，它和 rarp 现在很少使用，基本上被 bootp 和 dhcp 取代了。

chargen
tcp 版本的 chargen server
预设端口：TCP/UDP 19
说明：Character Generator Protocol，一种网络服务，主要功能是提供类似远程打字的功能。
是否需要启动：为安全起见，尽量关闭这个服务。

chargen-udp
udp 版本的 chargen server

cpuspeed
说明：监测系统空闲百分比，降低或加快 CPU 时钟速度和电压从而在系统空闲时将能源消耗降为最小，而在系统繁忙时最大化加快系统执行速度。
是否需要启动：需要启动。

crond
cron 是 Unix 下的一个传统程序，该程序周期地运行用户调度的任务。比起传统的 Unix 版本，Vixie 版本添加了不少属性，而且更安全，配置更简单。
配置文件：/etc/crontab
预设端口：无
说明：用来执行例行性命令的守护程序。
是否需要启动：必须启动。

cups
配置文件：
CUPS 服务器配置文件：/etc/cups/cupsd.conf
CUPS 客户端配置文件：/etc/cups/client.conf
CUPS 打印机配置文件：/etc/cups/printers.conf
CUPS 中类（class）配置文件：/etc/cups/classes.conf
说明：Common UNIX Printing System，公共 UNIX 打印支持，为 Linux 提供打印功能。
是否需要启动：如果不安装打印机，就不需要启动。

cups-lpd
预设端口：无
说明：CUPS Line Printer Daemon ("LPD")，提供打印功能。
是否需要启动：如果不安装打印机，就不需要启动。

daytime
tcp 版本的 daytime server
预设端口：TCP 13
说明：Daytime 协议（RFC867）是一个简单的协议，为客户机实现从远程服务器获取日期和时间的功能。

是否需要启动：不用启动。

daytime-udp
udp 版本的 daytime server

dhcpd
该守护提供了对动态主机控制协议(Dynamic Host Control Protocol)的访问支持。

echo
tcp 版本的 echo server
预设端口：7
说明：服务器回显客户数据服务。
是否需要启动：不用启动。

echo-udp
udp 版本的 echo server

eklogin
接受 rlogin 会话鉴证和用 kerberos5 加密的一种服务

finger
用于应答 finger 请求的服务

gated
gated 通过一个数据库提供了网络路由功能支持。它支持各种路由协议，包括 RIP 版本 1 和 2、DCN HELLO 协议、OSPF 版本 2 以及 EGP 版本 2 到 4。

gpm
gpm 为文本模式下的 Linux 程序如 mc(Midnight Commander)提供了鼠标的支持。它也支持控制台下鼠标 的拷贝，粘贴操作以及弹出式菜单。
配置文件：/etc/sysconfig/mouse
预设端口：无
说明：General Purpose Mouse Daemon ，gpm 为文本模式下的 Linux 程序如 mc(Midnight Commander)提供了鼠标的支持。它也支持控制台下鼠标 的拷贝，粘贴操作以及弹出式菜单。
是否需要启动：没必要的话，建议不要启动。

gssftp
接受可被 kerberos5 验证的 ftp 连接

httpd
http 是著名的 www 服务器，可用来提供 HTML 文件以及 CGI 动态内容服务。

identd
提供验证身份的方法,AUTH 服务，在提供用户信息方面与 finger 类似

inetd
因特网操作服务程序。监控网络对各种它管理的服务的需求，并在必要的时候启动相应的服务程序。通常，inetd 管理的程序有 telnet、ftp、rsh 和 rlogin。关闭 inetd 也就关闭了这些由它管理的服务。

innd
inn 是最流行的用户组新闻服务器。它允许您建立起本地新闻服务器。配置有一定的难度，可以先阅读/usr/doc/inn*文档获得帮助。

ipchains
ipchains 包过滤防火墙

iptables
iptables 包过滤防火墙
说明：防火墙。
是否需要启动：必须启动。

ipvsadm
调用 ipvsadm 来建立和维护 ipvs 路由选择表

irda
Infrared Data Association，是一个实现红外无线数据传输的工业标准。

irqbalance
对多个系统处理器环境下的系统中断请求进行负载平衡的守护程序。
是否需要启动：如果你只安装了一个 CPU，就不需要加载这个守护程序。

isdn
启用 isdn(综合服务数字网 Integrated Services Digital Network)服务.提供对 isdn 设备的支持。

kadmin
更改在主控 kdc 中使用本机的 kadmin 工具，或透过 kadmin 服务来完成

kdcrotate
设置配置文件/etc/krb5.con 中的 kdcs 表项

keytable
该程序的功能是转载您在/etc/sysconfig/keyboards 里说 明的键盘映射表，该表可以通过 kbdconfig 工具进行选择。您应该使该程序处于激活状态。

klogin
接受 bsd 方式的 rlogin 会话，但需要使用 kerbeos5 验证

kprop
是否允许 kdc 接收来的 master kdc 的升级

krb5-telnet
允许普通的 telnet 登陆，但也可使用 kerberos5 验证

krb524
是以改变 kerberos5 到 kerberosIV 的赁证

krb5kdc
开启 kerberosIV 和 5 所需的连接以获得赁证

kshell
接受 rshell 命令鉴证和用 kerberos 加密的服务

kudzu
运行硬件检测，并可选择性地设置硬件变化
配置文件：
/etc/sysconfig/hwconf
/etc/sysconfig/kudzu
说明：硬件自动检测程序，会自动检测硬件是否发生变动，并相应进行硬件的添加、删除工作。当系统启动时，kudzu 会对当前的硬件进行检测，并且和存储在 /etc/sysconfig/hwconf 中的硬件信息进行一一对照，如果某个硬件从系统中被添加或者删除时，那么 kudzu 就会察觉到，并且通知用户是否进行相关配置，然后修改/etc/sysconfig/hwconf，使硬件资料与系统保持同步。如果 /etc/sysconfig/hwconf 这个文件不存在，那么 kudzu 将会从/etc/modprobe.conf，/etc/sysconfig/network-scripts/和 /etc/X11/XF86Config 中探测已经存在的硬件。
是否需要启动：如果启动 kudzu，则每次启动系统，都会检查新硬件（checking new hardware），会延长系统启动的时间。如果你不打算增加新硬件，那么就可以关闭这个启动服务，以加快系统启动时间。

ldap
LDAP 代表 Lightweight Directory Access Protocol，实现了目录访问协议的行业标准。

linuxconf
linuxconf 是 Linux 下的一个有效的系统配置工具，该服务允许远程运行。

lpd
lpd 是系统打印守护程序，负责将 lpr 等程序提交给打印作业。

mars-nwe
mars-nwe 文件和用于 Novell 的打印服务器

mcserv
Midnight Commander 服务进程允许远程机器上的用户 通过 Midnight Commander 文件管理器操作本机文件。服 务进程用 PAM 来验证用户，需要给出“用户名/口令” 以通过验证。

mdmonitor
与 RAID 设备相关的守护程序。

mdmpd
与 RAID 设备相关的守护程序。

messagebus
D-BUS 是一个库，为两个或两个以上的应用程序提供一对一的通讯。
dbus-daemon-1 是一个应用程序，它使用这个库来实现 messagebus 守护程序。多个应用程序通过连接 messagebus 守护程序可以实现与其他程序交换信息。

microcode_ctl
可以编码以及发送新的微代码到 kernel 以更新 Intel IA32 系列处理器（Pentium Pro，PII，PIII，Pentium 4，Celeron, Xeon 等等 - 全部 P6 以及更高,不包括 pentium classics）。

mysql
一个快速高效可靠的轻型 SQL 数据库引擎。

named
域名服务器，将 Internet 主机名解析为点分的 IP 地址。

netdump
News Backup Dump Server，远程备份服务器。

netfs
负责装载/卸载 NFS、Samba、NCP(Netware)文件系统。

netplugd
配置文件：
/etc/netplug/netplugd.conf
/etc/netplug.d/netplug
说明：network cable hotplug management daemon，netplugd 是一个守护程序，可以监控一个或多个网络接口的状态，当某些事件触发时运行一个外部脚本程序。

network
激活/关闭启动时的各个网络接口。

nfs
NFS 是一个流行的基于 TCP/IP 网络的文件共享协议。该服务提供了 NFS 文件共享服务，具体的配置在/etc/exports 文件里。

nfslock
提供 NFS 文件锁定功能

nscd
nscd(Name Switch Cache daemon)服务器，用于 NIS 的一个支持服务，它高速缓存用户口令和组成成员关系.该服务负责密码和组的查询，并且缓冲查询结果。如果您的系统有比较慢的服务(如 NIS 和 NIS+)，则应该启动该服务。

ntpd
配置文件：/etc/ntp.conf
说明：Network time Protocol daemon，网络时间校正协议。简单的说,NTP 是用来使系统和一个精确的时间源保持时间同步的协议。

pcmcia
支持笔记本电脑的 PCMCIA 设备，如调制解调器, 网络适配器, SCSI 卡等等。

portmap
用来支持 RPC 连接，RPC 被用于 NFS 以及 NIS 等服务。如 NIS 和 NFS 提供动态端口的分配。

postgresql
关系数据库引擎

proftpd
是 Unix 下的一个配置灵活的 ftp 守护程序。

psacct
包括几个工具用来监控进程活动的工具，包括 ac,lastcomm, accton 和 sa。

radvd
路由广播程序。

random
保存和恢复系统的高质量随机数生成器，这些随机数是系统一些随机行为提供的。
说明：快速的将系统的状态在随机的时间内存到景象文件中，对于系统相当重要。因为在开机之后，系统会迅速的恢复到开机之前的状态。

是否需要启动：必须启动。

rawdevices
在使用集群文件系统时用于加载 raw 设备的守护程序。

readahead

readahead_early
配置文件：/etc/readahead.early.files /etc/readahead.files
说明：readahead 和 readahead_early 是在 Fedora core 2 中最新推出的两个后台运行的守护程序。其作用是在启动系统期间，将启动系统所要用到的文件首先读取到内存中，然后在内存中进行执行，以加快系统的启动速度。而上面两个配置文件就保存着将要读取到内存的文件列表。

rhnsd
Red Hat 网络服务。通知你有关官方的安全信息以及为你的系统打补丁。

rlogin
rlogin 程序服务，提供来自远程信任主机的注册功能

routed
该守护程序支持 RIP 协议的自动 IP 路由表维护。RIP 主要使用在小型网络上，大一点的网络就需要复杂一点的协议。

rpcgssd

rpcidmapd

rpcsrcgssd
说明：gestion NFS v4，是 Linux 2.6 内核新添的功能。
是否需要启动：不需要启动。

rsh
提供 rcmd 程序或者 rsh 程序的服务

rstatd
Rstat 协议允许网络上的用户获得同一网络上各机器的性能参数。

rsync
remote sync，远程数据备份工具。对 ftp 服务的一个很好的附加，允许循环码求和校验等

ruserd

远程用户定位服务，这是一个基于 RPC 的服务，该服务使网络用户可以定位同一网络上的其他用户。

rwall
激活 rpc.rwall 服务进程，这是一项基于 RPC 的服务，允许用户给每个注册到 LAN 机器上的其他终端写消息.Rwall 协议允许远程用户向在同一系统中活跃着的终端发送消息，类似 wall 的本地行为。

rwhod
激活 rwhod 服务进程，它支持 LAN 的 rwho 和 ruptime 服务,允许远程用户获得运行 rwho 守护的机器上所有已登录用户的列表，与 finger 类似。

saslauthd
使用 SASL 的认证守护程序。

sendmail
邮件服务器 sendmail

services
一个内部 xinetd 服务，用于监听活动的服务。

sgi-fam
实现实时数据镜像。监控文件的变更，提供一个应用程序 API 接口用来当指定的文件火目录改变时及时通知。

smartd
Self Monitor Analysis and Reporting Technology System，监控你的硬盘是否出现故障。

smb
Samba 文件共享/打印服务

snmpd
简单网络管理协议(SNMP)的守护

snortd
一个轻量级的网络入侵检测工具

squid
代理服务器 squid

sshd
OpenSSH 服务器

配置文件：
OpenSSH 服务器配置文件：/etc/ssh/sshd_config
OpenSSH 客户端配置文件：/etc/ssh/ssh_config
预设端口：22
说明：Secure Shell Protocol，实现安全地远程登陆管理主机。
是否需要启动：如果想实现远程管理，就需要启动。

swat
samba 网络配置工具，可以通过流览器的 901 端口连接使用 swat

syslog
syslog 是操作系统提供的一种机制，守护程序通常使用这种机制将各种信息写到各个系统日志文件。通常应该启动该服务。让系统引导时启动 syslog 和 klogd 系统日志守护
配置文件：/etc/syslog.conf
说明：记录所有的系统行为。
是否需要启动：必须启动。

telnet
提供 telnet 服务，使用未加密的用户/密码组进行验证

time
tcp 版本的 rfc 868 time server
从远程主机获取时间和日期，采用 TCP 协议。

time-udp
udp 版本的 rfc 868 time server
从远程主机获取时间和日期，采用 UDP 协议。

vncserver
VNC （Virtual Network Computing，虚拟网络计算），它提供了一种在本地系统上显示远程计算机整个"桌面"的轻量型协议。

webmin
webmin 是基于 web 的集系统管理与网络管理于一身的强大管理工具

xfs
X Window 字型服务器，为本地和远程 X 服务器提供字型集
预设端口：TCP 7100
说明：x font server，X Window 字型服务器，为本地和远程 X 服务器提供字型集。
是否需要启动：如果使用 run-level 为 5 的图形界面，那么就需要启动。

xinetd

因特网操作服务程序。提供类似于 inetd+tcp_wrapper 的功能，但是更加强大和安全，监控网络对各种它管理的服务的需求，并在要的时候启动相应的服务程序
配置文件：/etc/xinetd.conf
说明：xinetd 作为 inetd 的后续版本，负责管理系统中不频繁使用的服务，这些服务程序在有请求时才由 xinetd 服务负责启动运行，一旦完成服务请求服务程序结束运行，这样可以有效地减少对系统资源的占用率。通常，xinetd 管理的程序有 telnet、ftp、rsh 和 rlogin。关闭 inetd 也就关闭了这些由它管理的服务。
是否需要启动：必须启动。

xntpd
网络时间服务器

ypbind
为 NIS（网络信息系统）客户机激活 ypbind 服务进程

yppasswdd
NIS 口令服务器,让 NIS 用户能够修改密码。运行在 NIS 域的服务器上。客户端程序同样也叫 yppasswd。

ypserv
NIS 主服务器,标准 NIS/YP 网络协议的一个实现。允许主机名，用户名和其他信息分布于网络各端。运行在 NIS 服务器上，客户端不需要。

yum
配置文件：/etc/yum.conf
说明：Yellow Dog UpdaterModified，是一个自动更新、安装和删除 RPM 软件包的管理程序，它会自动计算软件包的管理程序，并判断哪些软件应该安装，哪些软件则不必安装。
是否需要启动：以系统管理策略而决定是否启动。

42 ☆登录过程

getty 输出 /etc/issue 文件中的内容，并显示登录
用户在 getty 的提示处输入登录名;
login 提示符由 getty 产生，并且在用户输入口令和密码时调用/bin/login
login 要求输入口令和密码，并对照/etc/shadow 验证口令
login 输出/etc/motd 中的每日消息，并且运行一个 shell;
shell 执行适当的引导脚本
shell 显示一个提示符，等待输入

RedHat 使用的是 mingetty 一种简化版本，只能处理虚拟终端的登录。而 mgetty 不仅可以处理虚拟终端，而且还能处理传真，支持调制解调器双向传输

43 ☆issue

/etc/issue
更改系统登录标题

1. 我们将设置 rc.local 脚本用于每次重启时出现登录标题. 打开/etc/rc.local 文件找到以下行:
touch /var/lock/subsys/local
2. 在后面插入以下行:
echo " Welcome to \n" > /etc/issue
echo "All access to this computer is monitored" >> /etc/issue
echo "Unauthorized access is prohibited" >> /etc/issue
echo >> /etc/issue
echo "Last reboot complete at $(/bin/date)" >> /etc/issue
3. 保存文件,把/etc/issue 复制为/etc/issue.old
4. 重启动系统
5. 当系统启动后,切换到虚拟控制台确认登录标题出现了. 打开/etc/issue, 注意 mingetty 把\n 扩展为你的主机名

44 ☆motd

/etc/motd
增加当天的消息

1. 编辑/etc/motd 文件,默认应为空. 增加以下行:
##################################
Welcome to station xx
##################################
<date> The sysadmin is playing today.
Expect frequent system downtime.
2. 切换到虚拟控制台登录.

45 ☆Linux 内核

内核官方
http://www.kernel.org

我们知道 Linus Torvalds 开发了 Linux ，其实他开发的就是内核，按内核官方主页的理解，这个内核就是 Linux ；其它的扩展和应用都是围绕内核而展开的。所有 Linux 应用程序都会和内核发生直接或者间接的接触；比如硬件需要内核支持，网络的通信也需要内核支持；文件系统更需要内核支持... ...

46 ☆为什么需要编译内核和管理内核

硬件是需要内核支持才行，有些硬件的支持没有被编入内核，这也需要我们重编内核；内核的包含的不仅仅是设备的驱动,还有其它的内容,比如网络协议的支持,防火墙的支持... ... 比如 iptables 的实现，有些功能是需要内核支持的，如果内核与 iptables 相关的内容没有被编入，iptables 相关的功能就无法实现；

47 ☆lsmod

lsmod 是列出目前系统中已加载的模块的名称及大小等；另外我们还可以查看 /proc/modules ，我们一样可以知道系统已经加载的模块；

48 ☆modinfo

modinfo 可以查看模块的信息，通过查看模块信息来判定这个模块的用途；

moinfo 模块名
模块名是不能带有后缀的，我们通过 modprobe -l 所看到的模块，都是带有.ko 或.o 后缀；

49 ☆rmmod

移除已挂载模块

rmmod 模块名
模块名是不能带有后缀的，我们通过 modprobe -l 所看到的模块，都是带有.ko 或.o 后缀；

50 ☆insmod

insmod 这个工具是挂载模块的，和 modprobe 有点类似，但功能上没有 modprobe 强，modprobe 在挂载模块是不用指定模块文件的路径，也不用带文件的后缀.o 或.ko ；而 insmod 需要的是模块的所在目录的绝对路径，并且一定要带有模块文件名后缀的

模块的路径用 modprobe -l 查看

不推荐使用。因为模块有依赖关系，对于新手来说，可能不知道这个模块依赖和哪个模块依赖；

insmod /lib/modules/2.6.11-1.1369_FC4/kernel/drivers/net/tg3.ko
我们要到 /lib/modules/内核版本 "uname -r" 的命令输出/kernel/drivers 中找相对应的模块才行，要有绝对路径，而且必须要用到文件名的全称，不能把文件名的后缀省略；

51 ☆modprobe

挂载新模块以及新模块相依赖的模块
modprobe 我们常用的功能就是挂载模块，在挂载某个内核模块的同时，这个模块所依赖的模块也被同时挂载；当然 modprobe 也有列出内核所有模块，还有移除模块的功能；

modprobe [-v] [-V] [-C config-file] [-n] [-i] [-q] [-o <modname>] <modname> [parameters...]
modprobe -r [-n] [-i] [-v] <modulename> ...
modprobe -l -t <dirname> [-a <modulename> ...]

modprobe -c
可以查看 modules 的配置文件，比如模块的别名是什么等

modprobe -l
是列出内核中所有的模块(以.ko 或.o 结尾的)，包括已挂载和未挂载的；通过 modprobe -l ，我们能查看到我们所需要的模块，然后根据我们的需要来挂载；其实 modprobe -l 读取的模块列表就位于 /lib/modules/'uname -r' 目录中；其中 uname -r 是内核的版本

挂载一个模块
modprobe 模块名
模块名是不能带有后缀的，我们通过 modprobe -l 所看到的模块，都是带有.ko 或.o 后缀；

移除已加载的模块
modprobe -r 模块名
和 rmmod 功能相同；

52 ☆depmod

depmod — program to generate modules.dep and map files.
创建模块依赖关系的列表
这个模块管理工具是创建模块依赖关系的列表

depmod -a
为所有列在/etc/modprobe.conf 或/etc/modules.conf 中的所有模块创建依赖关系，并且写入到 modules.dep 文件；

depmod -e
列出已挂载但不可用的模块；

depmod -n
列出所有模块的依赖关系，但仅仅是输出出来 （Write the dependency file on stdout only）

modules.dep 位于 /lib/modules/内核版本 目录

53 ☆配置内核需要的软件包

modutils 工具包
mkinitrd 程序包依赖于 device-mapper 包
mkinitrd-4.1.18-2

54 ☆modprobe.conf

内核模块的配置文件
modules.conf 或 modprobe.conf

内核模块的开机自动挂载模块一般是位于一个配置文件，一般的 Linux 发行版本都有 /etc/modules.conf 或 /etc/modprobe.conf 。

RHEL 下是/etc/modprobe.conf

比如 Fedora Core 4.0 内核模块开机自动加载文件是 /etc/modprobe.conf；在这个文件中，一般是写入模块的加载命令或模块的别名的定义等；比如我们在 modules.conf 中可能会发行类似的一行；
alias eth0 8139too

这样系统启动的时候，首先会 modprobe 8139too(插入模块)，然后再为 8139too 指定别名为 eth0，然后我们在登录的时候，用 ifconfig 就会查看到网卡的 IP 等情况，当然您得为网卡设置 IP 才行；

一般的情况下，modproe.conf 或 modules.conf 的内容是我们用相应的硬件配置工具而生成的；如果您的硬件驱动是没有被内核支持，您自己到硬件的厂商下载而来的驱动。一般的情况下都有安装和帮助文件。他们的驱动在配置时，他会写入硬件的支持到 modules.conf 或 modprobe.conf 文件中。

再比如我们的声卡在 modules.conf 或 modprobe.conf 中也有相应的内容，这是由 alsaconf 配置工具生成的，明白了吧；同理网卡在 modprobe.conf 或 modules.conf 中的内容也是由网卡的配置工具而来的。

有些硬件是以内核模块的方式驱动的，模块一旦加载上就能用，也没有什么配置工具，比如 vfat 和 ntfs 的支持；如果是硬件驱动不以模块的方式支持，而是直接编入内核，也不会用在 modprobe.conf 或 modules.conf 中加入什么内容；

!!!!!如果您有些模块不能开机加载，您想让一些模块加机自动加载，就可以把"modprobe 模块" 直接写入配置文件；

内核模块的其它配置文件还是需要了解的，比如 /lib/modules/内核版本目录下的几个文件；了解一下就行；

[root@panda pam.d]# uname -r
2.6.9-5.EL
[root@panda pam.d]# ls /lib/modules/2.6.9-5.EL/
build modules.alias modules.dep modules.inputmap modules.pcimap modules.usbmap
kernel modules.ccwmap modules.ieee1394map modules.isapnpmap modules.symbols source

/etc/modprobe.conf 文件
该配置文件定义了各种需要在启动时加载的模块的参数信息

语法
- alias Allows you to bind a name to a module.
- options Allows you to specify options for a module.
- install module command Use command instead of insmod on this module.
- pre-install module command Run command before installing this module.
- post-install module command Run command after installing this module.
- remove module command Use command instead of rmmod on this module.
- pre-remove module command Run command before loading this module.
- post-remove module command Run command after loading this module.

Linux 内核不会自动检测多个网卡。对于没有将网卡的驱动编译到内核而是作为模块动态载入的系统若需要安装多块网卡，应该在“modprobe.conf”文件中进行相应的配置。若设备驱动被编译为模块（内核的模块）：对于 PCI 设备，模块将自动检测到所有已经安装到系统上的设备;对于 ISA 卡，则需要向模块提供 IO 地址，以使模块知道在何处寻找该卡，这些信息在“/etc/modprobe.conf”中提供。

例如，我们有两块 ISA 总线的 3c509 卡，一个 IO 地址是 0x300，另一个是 0x320。编辑“modules.conf”文件如下：
alias eth0 3c509
alias eth1 3c509
options 3c509 io=0x300,0x320

对于 PCI 卡，仅仅需要 alias 命令来使 ethN 和适当的驱动模块名关联，PCI 卡的 IO 地址将会被自动的检测到。对于 PCI 卡，编辑"modprobe.conf"文件如下：
alias eth0 3c905
alias eth1 3c905

若驱动已经被编译进了内核：系统启动时的 PCI 检测程序将会自动找到所有相关的网卡。ISA 卡一般也能够被自动检测到，但是在某些情况下，ISA 卡仍然需要做下面的配置工作：在

"/etc/lilo.conf"中增加配置信息，其方法是通过 LILO 程序将启动参数信息传递给内核。对于 ISA 卡，编辑"lilo.conf"文件，增加如下内容：
append=" ether="0,0,eth0 ether="0,0,eth1"

55 ☆内核的硬件驱动模块

硬件驱动是必须由内核支持的，无论是我们自己安装驱动，还是内核自带的驱动都是如此。硬件驱动如果是以内核模块支持的，驱动目录位于： /lib/modules/内核版本/kernel/ 这个目录 或 /lib/modules/内核版本/kernel/drivers 目录中；

[root@panda pam.d]# uname -r
2.6.9-5.EL
[root@panda pam.d]# ls /lib/modules/2.6.9-5.EL/kernel/
arch crypto drivers fs lib net sound

只有驱动在内核中以模块的方法支持，驱动才位于 /lib/modules/相应的目录；如果是直接置入内核的，不会出现在/lib/modules 驱动相关的目录；

56 ☆编译内核的硬件驱动

通过源码编译驱动一般是./configure ;make;make install ，有时程序不提供./configure ，我们可以 make 或 make install ，或者执行 make;make install ；如果不能 make install ，则需要我们自己复制.o 或者.ko 文件到 /lib/modules/内核版本/kernel/ 目录 或 /lib/modules/内核版本/kernel/drivers 目录中相应的驱动目录；

具体情况的还是驱动的 REAME 和 INSTALL 为准；

现在大多驱动都是在编译安装时，都自动复制.o 或.ko 文件到内核模块目录，大多不用我们自己动手复制过去。如果您尝试编译安装声卡驱动 alsa-drivers 就会明白我所说的意思；

57 ☆内核的解压和安装

内核源码是放在 /usr/src/kernels 目录中；
如果通过在线升级内核，也是放在这个目录中；
如果您的系统中的 /usr/src/kernels/ 中没有内容，说明您没有安装内核的源码包 kernel-devel 软件包

如果您是通过在线安装的内核源码包 ，比如通过 apt+synaptic 或者 yum 安装的，内核源码会被放到/usr/src/kernel 下的目录中，您要进入相对应的目录进行编译；

如果您是是下载 kernel 和 kernel-devel 的 rpm 包，可以通过来安装；
[root@panda panda]# rpm -ivh kernel*.rpm

如果您是从 kernel.org 下载的类似 linux-2.6.13.tar.bz2 或者 linux-2.6.13.tar.gz 的，您要把下载下来的文件移到 /usr/src 目录中解压； 然后进入解压的目录中进行配置和编译；
[root@panda panda]# mv linux-2.6.13.tar.bz2
[root@panda panda]# cd /usr/src/
[root@panda src]# tar xvjpf linux-2.6.12.3.tar.bz2

58 ☆内核的配置

对于源码包
对于从 kernel.org 下载而来的 tar.bz 格式的源码包
进入目录执行 make mrproper
[root@panda src]# cd linux-2.6.12.3/
[root@panda linux-2.6.12.3]#
[root@panda src]# cd linux-2.6.12.3/
[root@panda linux-2.6.12.3]# make mrproper
[root@panda linux-2.6.12.3]# make menuconfig
make mrproper 是清理代码树的动作，保证源代码是干净的
编译过程中如果出错的话，就执行一下 make rmproper.这个命令是清除原来的*.o 文件的，但是如果你清除了他们以后，编译会非常费时间，因为这些 obj 文件需要重新生成。这样能解决一些编译过程的错误
（检查有无不正确的.o 文件和依赖关系，使用刚下载的完整的源程序包进行编译，所以本步可以省略。而如果你多次使用了这些源程序编译内核，那么最好要先运行一下这个命令。 ）

make menuconfig 如果失败，很可能是 ncurses 库没有装，可以用 make xconfig 或者 make config 来替代
make xconfig 是基于 QT 库
make gconfig 是基于 GTK 库.
它们好像还不太稳定

源代码中有 config-2.6.12.3 这个文件,复制成.config,在这个基础上 make menuconfig 更方便.

对于 RPM 包
如果您是通过在线安装的 kernel 和 kernel-devel 新版本的包，
比如是 2.6.12-1.1398_FC4-i686，你可以直进入 /usr/src/kernel/相应的目录中直接执行 make menuconfig ；
利用发行版本提供的 .config 来配置，这样方便点。
不要 make mrproper ，否则.config 就没有了；这也是为什么要用发行版本提供的内核源码升级包的原因；

59 ☆内核的编译

[root@panda linux-2.6.12.3]# make
如果不放心，也可以用旧的命令：make bzImage && make modules
以前的版本中会在 make bzImage && make modules 之前,用到 make dep

60 ☆内核的安装

[root@panda linux-2.6.12.3]# make modules_install
如果不放心，也可以用 cp ./bzImage /boot/vmlinuz-2.6.12.3 && cp System.map /boot/System.map-2.6.12.3 来替代
另外， make modules_install 也一样还可以用。
不过，建议在没有 make clean 之前，如果增加了新的模块，可以直接 make menuconfig && make moduels && make modules_install，不需要全部重新来过

这样就编译好了，并把模块也安装在了 /lib/modules 目录中了，请看：
[root@panda linux-2.6.12.3]# ls /lib/modules/
2.6.9-5.EL 2.6.12.3

61 ☆内核的镜像文件

创建 initrd 文件(一定要手工完成)
/sbin/mkinitrd initrd-2.6.12.3.img 2.6.12.3
不创建这个文件，有时是启动不起来的，比如提示 VFS 错误等

如果你的 kernel 支持内存镜像，就用 mkinitrd 命令制作一个镜像文件，然后拷贝到 /boot 路径下。
如果你要这么做，也要记得在 make menuconfig 的时候记得选择内核镜像 ram image，并且不能安装为模块，否则 initrd 就不会运作。

一.在内核配置时需要将“RAM Disk support”和“Initial RAM disk”编译进内核，同时将 SCSI 驱动编译成可加载模块，这样在 make install 后就可以生成 initrd.img 文件。
二.有时候即便如此 install 脚本也会弹出 No modules BusLogic found for kernel X 的错误，原因是 mdinitrd 程序没有找到已经编译好的 SCSI 驱动模块，从某种程度来说这是 install 脚本编写者所犯下的错误，为了弥补这种错误我们需要手工将 mkinitrd 所需要的可加载模块放在一个它能找到的位置，比如/lib/modules/2.x/scs/

62 ☆mkinitrd

mkinitrd -f -v /boot/initrd-$(uname -r).img $(uname -r)
制作系统的初始化映象

63 ☆内核的检验

ls /boot
如果看到了 vmlinuz-2.6.12.3 和 System.map-2.6.12.3 ，那么恭喜你，成功了！
如果你是用 make install 安装的，还会看到个 config-2.6.12.3 文件

64 ☆内核的清理

make clean
如果你以后还要利用这次编译的成果，也可以省略这一步，如果你想直接删除源代码目录，也可以省略这一步。

65 ☆修改 GRUB

在 grub 的配置文件中进行修改，增加新内核的支持。建议保留旧内核的项目，避免编译失败后无法启动。
通常 grub 配置文件在下面三个地方（根据发行版的不同）：
/etc/grub.conf
/boot/grub/menu.lst
/boot/grub/grub.conf
根据你的实际情况来修改。

如果重新启动后出现 kernel panic 错误或者显示应该修改 init 信息，则基本都是 kernel 语句错误。

66 ☆内核编译总结

make mrproper
make menuconfig
make
make install (make modules_install 可选)
make clean

旧版本中使用(EL3)

make mrproper
清除源代码

拷贝一个标准的配置(/boot/config-X 或/usr/src/linux-X/.config(make oldconfig 建立))到.config

定制配置
make config

make menuconfig
make xconfig
都可以 ,一样的

make dep
创建依赖性

make clean
清理源代码

make bzImage
建立内核图像

make modules
建立模块

make modules_install

67 ☆内核的配置菜单

内核配置有两种方法，一种是直接置入内核[*]；另一种是编成模块[M]；两种方法各有优点；直接编入内核的，比如设备的启动，不再需要加载模块的这一过程了；而编译成模块，则需要加载设备的内核支持的模块；但直接把所有的东西都编入内核也不是可行的，内核体积会变大，系统负载也会过重。我们编内核时最好把极为重要的编入内核；其它的如果您不明白的，最好用默认。
1）移动键盘上下左右键，按 Enter 进入一个目录。把指针移动到 Exit 就退出当前目录到上级目录；
2）针对自己机器存在的问题进行修改，比如大内存的支持；

先选中配置文件
选中 Load an Alternate Configuration File 项，把.config 调进来方便我们配置；

配置菜单如下:

Code maturity level options (代码成熟等级)

Code maturity level options ---->
［*］ Prompt for development and/or incomplete code/drivers
［*］ Select only drivers expected to compile cleanly

1.prompt for development and/or incomplete code/drivers.
默认情况下是选择的，这将会在设置界面中显示还在开发或者还没有完成的代码与驱动.你应该选择它，因为有许多设备可能必需选择这个选项才能进行配置，实际上它是安全的。

2.Select only drivers expected to compile cleanly(NEW)
选择这个选项你将不会看到一些已知的存在问题的驱动程序选项，默认的情况下也是选择的。如果 你有设备没有找到驱动选项，你可以将这一项去掉，或许就可以找到相关驱动了，不过它可能是有 BUG 的。

General setup (普通属性设置)

General setup ---->
（） Local version - append to kernel release
［*］ Support for paging of anonymous memory (swap)
［*］ System V IPC
［*］ POSIX Message Queues
［*］ BSD Process Accounting
［*］ BSD Process Accounting version 3 file format
［*］ Sysctl support
［ ］ Auditing support
（15） Kernel log buffer size (16 => 64KB, 17 => 128KB)
［*］ Support for hot-pluggable devices
［*］ Kernel Userspace Events
［*］ Kernel .config support
［*］ Configure standard kernel features (for small systems) --->

Support for paging of anonymous memory (swap) (NEW)
这个选项将使你的内核支持虚拟内存，也就是让你的计算机好象拥有比实际内存更多 多的内存空 间用来执行很大的程序。默认是选择的。

System V IPC
为进程提供通信机制，这将使系统中各进程间有交换信息与保持同步的能力。有些程序只有在选 Y 的情况下才能运行，这里一定要选。

POSIX Message Queues
这是 POSIX 的消息队列，它同样是一种 IPC。建议你最好将它选上。

BSD Process Accounting
这是充许用户进程访问内核将账户信息写入文件中的。这通常被认为是个好主意，

Sysctl support
这个选项能不重新编译内核修改内核的某些参数和变量，如果你也选择了支持/proc，将能从 /proc/sys 存取可以影响内核的参数或变量。建议你最好将它选上。

Auditing support

审记支持，用于和内核的某些子模块同时工作，例如 SELinux。只有选择此项及它的子项，才能调用有关审记的系统调用。

Kernel log buffer size (16 => 64 KB 17 => 128 KB)
内核日志缓存的大小

Kernel Userspace Events
内核中分为系统区和用户区，这里系统区和用户区进行通讯的一种方式

Kernel .config support
将.config 配置信息保存在内核中，选上它及它的子项使得其它用户能从/proc 中得到内核的配置。

Configure standard kernel features (for small systems)
这是为了编译某些特殊的内核使用的，通常你可以不选择这一选项

Loadable module support (加载模块选项)

Loadable module support ---->
［*］ Enable loadable module support
［*］ Module unloading
［*］ Module versioning support (EXPERIMENTAL)
［ ］ Source checksum for all modules
［*］ Automatic kernel module loading

Enable loadable module support
很多人喜欢将全部功能、硬件支持一股脑的编进内核，而不是使用模块的方式。使用模块支持，你的系统能具有更好的可扩充性。还有一个原因就是自己编写的功能模块、设备驱动模块（假设编写的质量不高）以模块方式工作引起 Kernel Panic 的机率要远远低于不支持模块全部编进内核的方式。

Module unloading
不选这个功能，加载的模块就不能卸载。没什么需要多解释的，建议最好选上。

Module versioning support (EXPERIMENTAL)
这个功能可以让你使用其它版本的内核模块

Source checksum for all modules
这个功能是为了防止更改了内核模块的代码但忘记更改版本号而造成版本冲突.

Automatic kernel module loading

这个选项能让内核自动的加载部份模块，建议你最好选上。举个例子说明一下，如模块 eth1394 依赖于模块 ieee1394。如果选择了这个选项，可以直接加载模块 eth1394；如果没有选择这个选项，必需先加载模块 ieee1394，再加载模块 eth1394，否则将出错。

Processor type and features (处理器内型及特性)

Subarchitecture Type (PC-compatible) --->
Processor family (386) --->
［ ］ Generic x86 support
［*］ HPET Timer Support
［*］ Symmetric multi-processing support
［ ］ Preemptible Kernel
［ ］ local APIC support on uniprocessors
［ ］ Machine Check Exception
［ ］ Check for non-fatal errors on AMD Athlon/Duron / Intel Pentium
＜M＞ Toshiba Laptop support
＜M＞ Dell laptop support
＜ ＞ /dev/cpu/microcode - Intel IA32 CPU microcode support
＜ ＞ /dev/cpu/*/msr - Model-specific register support
＜ ＞ /dev/cpu/*/cpuid - CPU information support
Firmware Drivers --->
High Memory Support (off) --->
［ ］ Math emulation
［*］ MTRR (Memory Type Range Register) support
［ ］ Use register arguments (EXPERIMENTAL)

Subarchitecture Type
这没什么好说的，如果用 PC 机的话都选这个。

Processor family (386)
这也没什么好说的，选择你机器对应的处理器即可。

Generic x86 support
这一选项针对 x86 系列的 CPU 使用更多的常规优化。如果你在上面一项选的是 i386、i586 之类的才选这个。

HPET Timer Support
HPET 是替代 8254 芯片的下一代时钟处理器。这里你可以安全的选上这一选项。如果硬件不支持的话，将仍使用 8254 时钟处理器。

Symmetric multi-processing support

对称多处理器支持，在单 CPU 的机器上，不选这个选项会更快一些。由于超线程技术，看起来是两颗 CPU，因些要选上这个选项。

Preemptible Kernel
这个选项能使应用程序即使内核在高负载时也很可靠，建议最好选上。

local APIC support on uniprocessors

Machine Check Exception
这个选项能让 CPU 检测到系统故障时通知内核

Check for non-fatal errors on AMD Athlon/Duron / Intel Pentium
打开这个选项将会检查你机器上可能存在的问题，如果有一个非致命错误出现将会自动的修复并且记录，这可以帮助你查出程序出现问题的原因，是一个不错的选项.

Toshiba Laptop support
对 Toshiba 本本的支持

Dell laptop suppor
对 Dell 的支持

/dev/cpu/microcode - Intel IA32 CPU microcode support
这个选项是让你使用不随 Linux 内核发行的 IA32 microcode，但是你必需有 IA32 microcode 的二进制文件。

/dev/cpu/*/msr - Model-specific register support
这个选项能让特权 CPU 访问 x86 的 MSR 寄存器。由于超线程并不是真正的多处理器环境，所以不要选择这个。

/dev/cpu/*/cpuid - CPU information support
这个选项能从/dev/cpu/x/cpuid 获得 CPU 的唯一标识符

High Memory Support (off)
如果你有大容量的内存(超过 4G)你要选它，以使内核可以使用这部分内存。偶是没这命啦这部分永远为 OFF，如果你有你就 ON 吧。

Math emulation
估计现在没人有 386 或 486SX 的处理器了吧，那就不要选这个。

MTRR (Memory Type Range Register) support
在 Intel p6 家族的处理器中(Ppro、 PII 和更新的)有一个内存类型范围寄存器，可用来控制处理器访问的内存范围。打开它一般可以提升显卡的显示性能

Power management options (ACPI, APM) (电源管理)

[*] Power Management support
[ ] Power Management Debug Support
[ ] Software Suspend (EXPERIMENTAL)
    ACPI (Advanced Configuration and Power Interface) Support --->
    APM (Advanced Power Management) BIOS Support --->
    CPU Frequency scaling --->

Power Management support
电源管理没什么好说的，不想浪费电就选上。如果不选你可以跳过这部份。

Power Management Debug Support
电源管理的调试信息支持，如果不是要调试内核有关电源管理部份，请不要选择这项。

Software Suspend (EXPERIMENTAL)
休眠到硬盘。也就是将内存写入交换分区中，下次启动可以通过参数 resume=/dev/swappartition（例如：resume= /dev/hda6）来恢复上次机器运行的状态。这项功能对于系统引导时启动许多服务的机器来说很有用，可以节约启动时间。这项功能根据自己的需要选择吧，如果你选择这项功能，记得恢复休眠后重做交换分区。

ACPI (Advanced Configuration and Power Interface) Support --->
从这里进入 ACPI 电源管理的配置界面，要注意 ACPI 与 APM 不能同时使用，如果你同时配置了这两者，那么在系统启动时如果发现一个可工作的 ACPI 设备那么 APM 将被关闭，ACPI 会被加载

APM (Advanced Power Management) BIOS Support --->
高级电源管理的支持，一般来说笔记本应该选上，台式机可以不选。

CPU Frequency scaling --->
这一选项允许改变 CPU 的主频，使 CPU 在低负荷或使用电池时降低主频，达到省电的目的。

Bus options (PCI, PCMCIA, EISA, MCA, ISA)

[*] PCI support
      PCI access mode (Any) --->
[ ] PCI Express support
[ ] Legacy /proc/pci interface
[ ] PCI device name database
[ ] ISA support
[ ] EISA support

[ ] Vesa Local Bus priming
[ ] Generic PCI/EISA bridge
[ ] EISA virtual root device
[ ] EISA device name database
[ ] MCA support
[ ] Legacy MCA API Support
[ ] Support For the mca entry in /proc
[ ] NatSemi SCx200 support
PCCARD (PCMCIA/CardBus) support ---->
PCI Hotplug Support ---->

PCI support
PCI 支持,如果使用了 PCI 插槽，当然必选

PCI access mode (Any)
选 Any，系统将优先使用 MMConfig，然后使用 BIOS，最后使用 Direct 检测 PCI 设备

PCI Express support
PCI Express 总线支持

Legacy /proc/pci interface
是否使用/proc/pci 目录下的信息文件来描述 PCI 设备的信息。现在的系统多数都使用 lspci 工具来得到这样的信息

PCI device name database
如果你不打算使用 lspci 工具，就把这项和上面的一项选上。lspci 和 hotplug 都不需要内核中的设备信息库了

ISA support
如果你没有老式的 ISA 设备,老的 ISA 槽支持,可以不选这项

EISA suppor
扩展工业总线支持

Vesa Local Bus priming
ESA 总线,也是扩展工业总线的一种,已经被 PCI 代替,所以不用选择

Generic PCI/EISA bridge
PCI、EISA 两种总线的桥

EISA virtual root device

EISA 总线的虚拟根设备

EISA device name database
内核中的 EISA 设备信息库

MCA support
(IBM 的东东) 微通道总线.IBM 的台式机和笔记本上可能会有这种总线,包括它的 p 系列、e 系列、z 系列机器上都用到了这种总线

Legacy MCA API Support

Support For the mca entry in /proc

NatSemi SCx200 support
松下的一种半导体处理器的驱动,If you don't know what to do here, say N.

PCCARD (PCMCIA/CardBus) support --->
一般只有笔记本电脑上才会有 PCMCIA 插槽,如果你是台式机的话,可以不选这一项

PCI Hotplug Support --->
支持 PCI 热插拔的.

Executable file formats (可执行文件格式)

[ ] Kernel support for ELF binaries
    [ ] Kernel support for a.out and ECOFF binaries
    [ ] Kernel support for MISC binaries

Kernel support for ELF binaries
这个当然 y,因为目前 gcc2.7.0 以上的都有支持 ELF 了,如果没有选择这一项可能会使用相当多的程序因此无法执行

Kernel support for a.out and ECOFF binaries
a.out 的执行文件是比较古老的可执行码,用在比较早期的 UNIX 系统上.Linux 最初也是使用这种码来执行程序,一直到 ELF 格式的可执行码出来后,有愈来俞多的程序码随着 ELF 格式的优点而变成了 ELF 的可执码.将来势必完全取代 a.out 格式的可执行码.但目前由于沿有许多的程序还没有取代过来,所以只好选择 Y,等将来有一天,全部的程序都变成了 ELF 的天下时,那时再 disable 掉

Kernel support for MISC binaries
可以让你支援别的种类的 binary 执行档(如：Java、Python ... etc)?G 到 kernel 或编成 module 都 ok

Device Drivers

[*] Generic Driver Options ---->
[ ] Memory Technology Devices (MTD) ---->
[ ] Parallel port support ---->
[ ] Plug and Play support ---->
[ ] Block devices ---->
Block devices 这一项
Normal floppy disk support (系统应该自己就选择了吧)
Parallel port IDE device
这一项，只选 了 Parallel port ATAPI CD-ROMs ,其它的不懂，就成了模块了
Loopback device support 忘了这是干么的了，从网上查，选 上好
Cryptoloop Support(这个是它的选项，选 上再说)
Network block device support (好像是网络支持还是)
Low Performance USB Block driver 不知道干么的，
RAM disk support (选 上吧，如果你有 usb,或者 scsi 就用到吧，俺选 了)
Packet writing on CD/DVD media

[ ] ATA/ATAPI/MFM/RLL support ---->
ATA/ATAPI/MFM/RL 这一项
Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
Include IDE/ATAPI CDROM support
IDE Taskfile Access
generic/default IDE chipset support
SCSI emulation support 这个不知道干么的，可能与俺的 scsi 有关就选 了
CMD640 chipset bugfix/support 系统就选 了
PCI IDE chipset support
Sharing PCI IDE interrupts support
Generic PCI IDE Chipset Support
Generic PCI bus-master DMA support
Intel PIIXn chipsets support
VIA82CXXX chipset support

[ ] SCSI device support ---->
[ ] Old CD-ROM drivers (not SCSI, not IDE) --->
[ ] Multi-device support (RAID and LVM) ---->
[ ] Fusion MPT device support ---->
[ ] IEEE 1394 (FireWire) support ---->
[ ] I2O device support ---->

[ ] Networking support --->
[ ] ISDN subsystem --->
[ ] Telephony support --->
[ ] Input device support --->
[ ] Character devices --->
[ ] I2C support --->
[ ] Dallas's 1-wire bus --->
[ ] Misc devices --->
[ ] Multimedia devices --->
[ ] Graphics support --->
[ ] Sound --->
[ ] USB support --->
[ ] MMC/SD card support --->
[ ] InfiniBand support --->

Generic Driver Options

Memory Technology Devices (MTD)

Parallel port support

Plug and Play support
热插拔支持，当然要选择.

Block devices

ATA/ATAPI/MFM/RLL support

SCSI device support

Old CD-ROM drivers (not SCSI, not IDE)

Multi-device support (RAID and LVM)

Fusion MPT device support

IEEE 1394 (FireWire) support

I2O device support

Networking support

ISDN subsystem

Telephony support

Input device support

Character devices

I2C support

Dallas's 1-wire bus

Misc devices

Multimedia devices

Graphics support

Sound

USB support

MMC/SD card support

InfiniBand support

68 ☆内核的配置菜单案例

选择自己机器的 CPU；
移动键盘到 Processor type and features ---> ，然后按 ENTER 进入；
找到 Processor family (Pentium-Pro) ---> 按 ENTER 进入；

进入后我们发现有好多 CPU 的型号可选；一般的情况下要根据
bash-3.00# cat /proc/cpuinfo 输出的信息来选.
当然默认的 486 也是可以正常运行的，既然我们重编一次内核，就得选中对应型号的，也许性能有所提高呢；

对大内存支持；如果内存是 1G 或者 1G 以上，但小于 4G 的，就要选 4G 支持；如果超过 4G 的，要选 64G 的支持；
High Memory Support (4GB) --->
(X) 4GB

( ) 64GB

还有比如声卡等硬件，需要我们一步一步的查看；如果有不明之处，就要按 [shift]+？的组合键来查看说明。一般的情况下，2.6.x 的内核会根据机器的情况自动配出一个文件，只需要我们来查看一下，把重要的地方改改就行了；

再举个例子：比如我现在所用的声卡是 intel ac97 的，我应该怎么配置呢？

首先要知道自己的声卡的芯片组，我们要通过 lspci -v 来查看；
[root@panda panda]#lspci -v
只查看声卡的，应该用如下的方法：
[root@panda panda]# lspci -v |grep audio
00:1f.5 Multimedia audio controller: Intel Corp. 82801DB (ICH4) AC'97 Audio Controller (rev 03)

通过上面的输出，我们知道这台机器用的是 intel AC97 声卡；所以我们要特别注意 AC97 的配置；
找到 Device Drivers ---> Sound --->
<M> Sound card support 声卡的支持，这个是一定要选中的吧；
<M> Advanced Linux Sound Architecture 对声卡支持的 ALSA 驱动的支持；
下面有 OSS 驱动，只是一部份。如果想用 OSS 的驱动更全的，可以去买；其它的就看如下的选吧；
<M> Sequencer support
<M> Sequencer dummy client
<M> OSS Mixer API
<M> OSS PCM (digital audio) API[*] OSS Sequencer API
<M> RTC Timer support[*] Verbose printk
[ ] Debug

大多是默认的就好，如果您不知道是做什么用的，或者怎么用；

Generic devices ---> 进入里面
<M> Dummy (/dev/null) soundcard
<M> Virtual MIDI soundcard
<M> MOTU MidiTimePiece AV multiport MIDI
<M> UART16550 serial MIDI driver
<M> Generic MPU-401 UART driver
ISA devices ---> 如果您用 ISA 声卡就在这里面选；
PCI devices ---> 如果您用 PCI 声卡，就在这里面选，集成声卡也在这里；
USB devices ---> 这是 USB 声卡内核支持选项；我有一个这样的声卡，但没有试过；
PCMCIA devices ---> 这是 PCMCIA 声卡的选项，我还没有看过这样的声卡呢；如果您有，就在这里面动动手吧。
因为我用的是 Intel 集成的声卡，所以要在 PCI 中选择，我们在 中可以看到有两个与 INTEL 有关的；

<M> Intel/SiS/nVidia/AMD/ALi AC97 Controller 这个才是 Intel AC97 声卡的；
< > Intel/SiS/nVidia/AMD MC97 Modem (EXPERIMENTAL) 这个是机器集成的 INTEL 猫的蜂鸣器的；
因为我发现如果把猫的蜂鸣器的驱动也选上，可能造成两个冲突。所以只能选上面的那个；

我们再回到 Open Sound System ---> 中看看，与我用的声卡是不是有关的？

代码:
<M> Open Sound System (DEPRECATED)
<M> Intel ICH (i8xx) audio support
<M> OSS sound modules
<M> Loopback MIDI device support
<M> Microsoft Sound System support
我们也可以看到 Open Sound System 中也有好多的声卡驱动，大家根据前面的 lspci -v 来选择吧。

对于操作系统所采用的文件系统的支持要编入内核，最好不要编成模块；（重要）

比如我的 Fedora core 4.0 所采用的文件系统用的是 ext3 ，所以我要把它直接编入内核；好处是不受模块丢失或者损坏而不能启动系统；而有时您把系统所采用的文件系统编译成模块，出现 VFS 错误，也有这方面的事，可能是您没有把 ext3 加入到相应的加载模块的配置文件中，所以我们为了减少麻烦，把风险降到最低，还是要直接置入内模的好；

File systems --->
<*> Ext3 journalling file system support[*] Ext3 extended attributes[*] Ext3 POSIX Access Control Lists[*] Ext3 Security Labels
如果您还有其它的硬盘分区要读取，比如是 reiserfs、ext2、fat、fat32、ntfs 等，这样的可以编成模块来支持；

再举一例：如果您的的操作系统用的是 reiserfs 的文件系统，当然就要把 reiserfs 的直接编入内核，其它的可以编成模块来支持了；

对于硬盘及 RAID 的支持，要直接编入内核；

比如 ATA、SATA、SCSI 及 RAID 的支持直接内核支持；有时编完内核后，启动时不能识别硬盘和 RAID，大多事情出在这里；Slackware 中在这方面有的是模块支持，我们可以把它由模块 M 改成内核*来支持； 如果您不明白，就按默认进行；SATA 的硬盘的支持除了选中 SATA 的支持、IDE 设备的支持以外，还要选中 SCSI 的支持；

对于咱们所没有的设备，可以在内核中不选，熟能生巧罢了；

比如我没有 ISDN 设备 ，所以就把 ISDN 去掉；

ISDN subsystem --->
< > Linux telephony support

特别是 Native Language Support（本地语言支持）这个，里面有个 chinease cp936
一定要选，否则 mount 光驱后加了 codepage=936,iocharset=cp936，中文还是显示为“??????”，

如果您没有 1394 的设备 ，当然可以把 1394 的支持也去掉；等等。。。。。。。

如果您有 USB 的设备，要把 USB 方面好好看看；比如大家常用的移动硬盘；USB 猫等，还有扫描仪等；

配置好后先要保存

Save Configuration to an Alternate File

出来一个

Enter a filename to which this configuration ， should be saved as an alternate. Leave blank to abort.
.config

按回车就行了，这样就保存住了；

然后退出 Exit ，这时也会出现保存 ；

如果你想把.config 保存起来，可以再复制一份到安全一点的目录，以备后用；

69 ☆Makefile

/usr/src/kernels/2.6.9-5.EL-i686/Makefile
是整个内核配置、编译的总体控制文件。

Makefile 的作用是根据配置的情况，构造出需要编译的源文件列表，然后分别编译，并把目标代码链接到一起，最终形成 Linux 内核二进制文件。

/usr/src/kernels/2.6.9-5.EL-i686/arch/*/Makefile
各种 CPU 体系目录下的 Makefile

对 Makefile 的说明和解释，由#开始。

Makefile 中的变量

顶层 Makefile 定义并向环境中输出了许多变量，为各个子目录下的 Makefile 传递一些信息。有些变量，比如 SUBDIRS，不仅在顶层 Makefile 中定义并且赋初值，而且在 arch/*/Makefile 还作了扩充。

常用的变量有以下几类：

1） 版本信息
版本信息有：VERSION，PATCHLEVEL, SUBLEVEL, EXTRAVERSION，KERNELRELEASE。
版本信息定义了当前内核的版本，比如 VERSION=2，PATCHLEVEL=4，SUBLEVEL=18，EXATAVERSION=-rmk7，它们共同构成内核的发行版本 KERNELRELEASE：2.4.18-rmk7

2） CPU 体系结构：ARCH
在顶层 Makefile 的开头，用 ARCH 定义目标 CPU 的体系结构，比如 ARCH:=arm 等。许多子目录的 Makefile 中，要根据 ARCH 的定义选择编译源文件的列表。

3） 路径信息：TOPDIR, SUBDIRS
TOPDIR 定义了 Linux 内核源代码所在的根目录。例如，各个子目录下的 Makefile 通过 $(TOPDIR)/Rules.make 就可以找到 Rules.make 的位置。
SUBDIRS 定义了一个目录列表，在编译内核或模块时，顶层 Makefile 就是根据 SUBDIRS 来决定进入哪些子目录。SUBDIRS 的值取决于内核的配置，在顶层 Makefile 中 SUBDIRS 赋值为 kernel drivers mm fs net ipc lib；根据内核的配置情况，在 arch/*/Makefile 中扩充了 SUBDIRS 的值，参见 4）中的例子。

4） 内核组成信息：HEAD, CORE_FILES, NETWORKS, DRIVERS, LIBS
Linux 内核文件 vmlinux 是由以下规则产生的：

```
vmlinux: $(CONFIGURATION) init/main.o init/version.o linuxsubdirs
$(LD) $(LINKFLAGS) $(HEAD) init/main.o init/version.o \
--start-group \
$(CORE_FILES) \
$(DRIVERS) \
$(NETWORKS) \
$(LIBS) \
--end-group \
-o vmlinux
```

可以看出，vmlinux 是由 HEAD、main.o、version.o、CORE_FILES、DRIVERS、NETWORKS 和 LIBS 组成的。这些变量（如 HEAD）都是用来定义连接生成 vmlinux 的目标文件和库文件列表。其中，HEAD 在 arch/*/Makefile 中定义，用来确定被最先链接进 vmlinux 的文件列表。比如，对于 ARM 系列的 CPU，HEAD 定义为：

```
HEAD := arch/arm/kernel/head-$(PROCESSOR).o \
```

arch/arm/kernel/init_task.o

表明 head-$(PROCESSOR).o 和 init_task.o 需要最先被链接到 vmlinux 中。PROCESSOR 为 armv 或 armo，取决于目标 CPU。 CORE_FILES，NETWORK，DRIVERS 和 LIBS 在顶层 Makefile 中定义，并且由 arch/*/Makefile 根据需要进行扩充。 CORE_FILES 对应着内核的核心文件，有 kernel/kernel.o，mm/mm.o，fs/fs.o，ipc/ipc.o，可以看出，这些是组成内核最为重要的文件。同时，arch/arm/Makefile 对 CORE_FILES 进行了扩充：

```
# arch/arm/Makefile

# If we have a machine-specific directory, then include it in the build.
MACHDIR := arch/arm/mach-$(MACHINE)
ifeq ($(MACHDIR),$(wildcard $(MACHDIR)))
SUBDIRS += $(MACHDIR)
CORE_FILES := $(MACHDIR)/$(MACHINE).o $(CORE_FILES)
endif

HEAD := arch/arm/kernel/head-$(PROCESSOR).o \
arch/arm/kernel/init_task.o
SUBDIRS += arch/arm/kernel arch/arm/mm arch/arm/lib arch/arm/nwfpe
CORE_FILES := arch/arm/kernel/kernel.o arch/arm/mm/mm.o $(CORE_FILES)
LIBS := arch/arm/lib/lib.a $(LIBS)
```

5） 编译信息：CPP, CC, AS, LD, AR，CFLAGS，LINKFLAGS
在 Rules.make 中定义的是编译的通用规则，具体到特定的场合，需要明确给出编译环境，编译环境就是在以上的变量中定义的。针对交叉编译的要求，定义了 CROSS_COMPILE。比如：

```
CROSS_COMPILE = arm-linux-
CC = $(CROSS_COMPILE)gcc
LD = $(CROSS_COMPILE)ld
  .....
```

CROSS_COMPILE 定义了交叉编译器前缀 arm-linux-，表明所有的交叉编译工具都是以 arm-linux- 开头的，所以在各个交叉编译器工具之前，都加入了 $(CROSS_COMPILE)，以组成一个完整的交叉编译工具文件名，比如 arm-linux-gcc。
CFLAGS 定义了传递给 C 编译器的参数。
LINKFLAGS 是链接生成 vmlinux 时，由链接器使用的参数。LINKFLAGS 在 arm/*/Makefile 中定义，比如：

```
# arch/arm/Makefile
```

LINKFLAGS :=-p -X -T arch/arm/vmlinux.lds

6） 配置变量 CONFIG_*
.config 文件中有许多的配置变量等式，用来说明用户配置的结果。例如 CONFIG_MODULES=y 表明用户选择了 Linux 内核的模块功能。
.config 被顶层 Makefile 包含后，就形成许多的配置变量，每个配置变量具有确定的值：y 表示本编译选项对应的内核代码被静态编译进 Linux 内核；m 表示本编译选项对应的内核代码被编译成模块；n 表示不选择此编译选项；如果根本就没有选择，那么配置变量的值为空。

70 ☆.config

/usr/src/kernels/2.6.9-5.EL-i686/.config
内核配置文件

```
#
# Automatically generated make config: don't edit
#
CONFIG_X86=y
CONFIG_MMU=y
CONFIG_UID16=y
CONFIG_GENERIC_ISA_DMA=y

#
# Code maturity level options
#
CONFIG_EXPERIMENTAL=y
CONFIG_CLEAN_COMPILE=y
CONFIG_STANDALONE=y
CONFIG_BROKEN_ON_SMP=y

# General setup
#
CONFIG_SWAP=y
CONFIG_SYSVIPC=y
CONFIG_POSIX_MQUEUE=y
# CONFIG_BSD_PROCESS_ACCT is not set
CONFIG_SYSCTL=y
CONFIG_AUDIT=y
CONFIG_AUDITSYSCALL=y
CONFIG_LOG_BUF_SHIFT=14
```

CONFIG_HOTPLUG=y
# CONFIG_IKCONFIG is not set
# CONFIG_EMBEDDED is not set
CONFIG_KALLSYMS=y
CONFIG_FUTEX=y
CONFIG_EPOLL=y
CONFIG_IOSCHED_NOOP=y
CONFIG_IOSCHED_AS=y
CONFIG_IOSCHED_DEADLINE=y
CONFIG_IOSCHED_CFQ=y
# CONFIG_CC_OPTIMIZE_FOR_SIZE is not set

在这里基本上是按默认选项，只是去掉了：CONFIG_BSD_PROCESS_ACCT is not set

Loadable module support
#
CONFIG_MODULES=y
CONFIG_MODULE_UNLOAD=y
# CONFIG_MODULE_FORCE_UNLOAD is not set
CONFIG_OBSOLETE_MODPARM=y
CONFIG_MODVERSIONS=y
CONFIG_KMOD=y
这些都要选择吧，如果你真的想把所有的东西都编辑到内核，那就不用考虑它了。

Processor type and features
#
CONFIG_X86_PC=y
# CONFIG_X86_ELAN is not set
# CONFIG_X86_VOYAGER is not set
# CONFIG_X86_NUMAQ is not set
# CONFIG_X86_SUMMIT is not set
# CONFIG_X86_BIGSMP is not set
# CONFIG_X86_VISWS is not set
# CONFIG_X86_GENERICARCH is not set
# CONFIG_X86_ES7000 is not set
# CONFIG_M386 is not set
# CONFIG_M486 is not set
# CONFIG_M586 is not set
# CONFIG_M586TSC is not set
# CONFIG_M586MMX is not set
# CONFIG_M686 is not set
# CONFIG_MPENTIUMII is not set

```
# CONFIG_MPENTIUMIII is not set
# CONFIG_MPENTIUMM is not set
CONFIG_MPENTIUM4=y
# CONFIG_MK6 is not set
# CONFIG_MK7 is not set
# CONFIG_MK8 is not set
# CONFIG_MCRUSOE is not set
# CONFIG_MWINCHIPC6 is not set
# CONFIG_MWINCHIP2 is not set
# CONFIG_MWINCHIP3D is not set
# CONFIG_MCYRIXIII is not set
# CONFIG_MVIAC3_2 is not set
# CONFIG_X86_GENERIC is not set
CONFIG_X86_CMPXCHG=y
CONFIG_X86_XADD=y
CONFIG_X86_L1_CACHE_SHIFT=7
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_GOOD_APIC=y
CONFIG_X86_INTEL_USERCOPY=y
CONFIG_X86_USE_PPRO_CHECKSUM=y
# CONFIG_HPET_TIMER is not set
# CONFIG_HPET_EMULATE_RTC is not set
# CONFIG_SMP is not set
# CONFIG_PREEMPT is not set
# CONFIG_X86_UP_APIC is not set
CONFIG_X86_TSC=y
CONFIG_X86_MCE=y
# CONFIG_X86_MCE_NONFATAL is not set
# CONFIG_TOSHIBA is not set
# CONFIG_I8K is not set
CONFIG_MICROCODE=m
CONFIG_X86_MSR=m
CONFIG_X86_CPUID=m


#
# Firmware Drivers
#
```

```
# CONFIG_EDD is not set
# CONFIG_SMBIOS is not set
CONFIG_NOHIGHMEM=y
# CONFIG_HIGHMEM4G is not set
# CONFIG_HIGHMEM64G is not set
# CONFIG_MATH_EMULATION is not set
# CONFIG_MTRR is not set
# CONFIG_EFI is not set
# CONFIG_REGPARM is not set
```

选择你的 CPU 吧，我的是:CONFIG_MPENTIUM4=y
另外对个 4G 的内存支持也不要了吧，CONFIG_NOHIGHMEM=y
这里要注意的还有：# CONFIG_REGPARM is not set 这一项呀，NO 吧

```
# Power management options (ACPI, APM)
#
CONFIG_PM=y
# CONFIG_SOFTWARE_SUSPEND is not set
# CONFIG_PM_DISK is not set

#
# ACPI (Advanced Configuration and Power Interface) Support
#
CONFIG_ACPI=y
CONFIG_ACPI_BOOT=y
CONFIG_ACPI_INTERPRETER=y
CONFIG_ACPI_SLEEP=y
CONFIG_ACPI_SLEEP_PROC_FS=y
CONFIG_ACPI_AC=m
CONFIG_ACPI_BATTERY=m
CONFIG_ACPI_BUTTON=m
CONFIG_ACPI_FAN=m
CONFIG_ACPI_PROCESSOR=m
CONFIG_ACPI_THERMAL=m
# CONFIG_ACPI_ASUS is not set
# CONFIG_ACPI_TOSHIBA is not set
# CONFIG_ACPI_DEBUG is not set
CONFIG_ACPI_BUS=y
CONFIG_ACPI_EC=y
CONFIG_ACPI_POWER=y
CONFIG_ACPI_PCI=y
CONFIG_ACPI_SYSTEM=y
```

```
CONFIG_X86_PM_TIMER=y

#
# APM (Advanced Power Management) BIOS Support
#
# CONFIG_APM is not set

#
# CPU Frequency scaling
#
# CONFIG_CPU_FREQ is not set
```

如果大家也是用 ACPI，就按这个选择试试吧\

```
# Bus options (PCI, PCMCIA, EISA, MCA, ISA)
#
CONFIG_PCI=y
# CONFIG_PCI_GOBIOS is not set
# CONFIG_PCI_GOMMCONFIG is not set
# CONFIG_PCI_GODIRECT is not set
CONFIG_PCI_GOANY=y
CONFIG_PCI_BIOS=y
CONFIG_PCI_DIRECT=y
CONFIG_PCI_MMCONFIG=y
CONFIG_PCI_LEGACY_PROC=y
# CONFIG_PCI_NAMES is not set
CONFIG_ISA=y
# CONFIG_EISA is not set
# CONFIG_MCA is not set
# CONFIG_SCx200 is not set

#
# PCMCIA/CardBus support
#
# CONFIG_PCMCIA is not set
CONFIG_PCMCIA_PROBE=y

#
# PCI Hotplug Support
#
CONFIG_HOTPLUG_PCI=m
# CONFIG_HOTPLUG_PCI_FAKE is not set
```

```
# CONFIG_HOTPLUG_PCI_COMPAQ is not set
# CONFIG_HOTPLUG_PCI_ACPI is not set
# CONFIG_HOTPLUG_PCI_CPCI is not set
# CONFIG_HOTPLUG_PCI_PCIE is not set
# CONFIG_HOTPLUG_PCI_SHPC is not set

#
# Executable file formats
#
CONFIG_BINFMT_ELF=y
# CONFIG_BINFMT_AOUT is not set
CONFIG_BINFMT_MISC=m
```

我不使用 PCMCIA 卡，所以选项就简单多了，当然 ISA 还是选上了，有啥用？我也不知道，真的，如果你想使用移动硬盘呀，闪存呀什么的，这里的选择可就要注意了。

```
#
# Device Drivers
#

#
# Generic Driver Options
#
# CONFIG_FW_LOADER is not set

#
# Memory Technology Devices (MTD)
#
# CONFIG_MTD is not set

#
# Parallel port support
#
# CONFIG_PARPORT is not set
```

这些配件都不想要，打印机我可以用 USB 的呀(我还真没看到过有谁的 NOTEBOOK 插着一个 PRINTER 的)，MTD 是什么东西？我没有，不要了。

```
# Plug and Play support
#
CONFIG_PNP=y
# CONFIG_PNP_DEBUG is not set
```

```
#
# Protocols
#
CONFIG_ISAPNP=y
# CONFIG_PNPBIOS is not set

#
# Block devices
#
CONFIG_BLK_DEV_FD=m
# CONFIG_BLK_DEV_XD is not set
# CONFIG_BLK_CPQ_DA is not set
# CONFIG_BLK_CPQ_CISS_DA is not set
# CONFIG_BLK_DEV_DAC960 is not set
# CONFIG_BLK_DEV_UMEM is not set
# CONFIG_BLK_DEV_LOOP is not set
# CONFIG_BLK_DEV_NBD is not set
# CONFIG_BLK_DEV_CARMEL is not set
# CONFIG_BLK_DEV_RAM is not set
# CONFIG_LBD is not set
```
看着选吧，如果你想用你的 MP3 的话(我这样说对不对呀？)

```
# ATA/ATAPI/MFM/RLL support
#
CONFIG_IDE=y
CONFIG_BLK_DEV_IDE=y

#
# Please see Documentation/ide.txt for help/info on IDE drives
#
# CONFIG_BLK_DEV_HD_IDE is not set
CONFIG_BLK_DEV_IDEDISK=y
CONFIG_IDEDISK_MULTI_MODE=y
# CONFIG_IDEDISK_STROKE is not set
CONFIG_BLK_DEV_IDECD=y
# CONFIG_BLK_DEV_IDETAPE is not set
# CONFIG_BLK_DEV_IDEFLOPPY is not set
# CONFIG_BLK_DEV_IDESCSI is not set
# CONFIG_IDE_TASK_IOCTL is not set
# CONFIG_IDE_TASKFILE_IO is not set
```

```
#
# IDE chipset support/bugfixes
#
CONFIG_IDE_GENERIC=y
# CONFIG_BLK_DEV_CMD640 is not set
CONFIG_BLK_DEV_IDEPNP=y
CONFIG_BLK_DEV_IDEPCI=y
CONFIG_IDEPCI_SHARE_IRQ=y
# CONFIG_BLK_DEV_OFFBOARD is not set
CONFIG_BLK_DEV_GENERIC=y
# CONFIG_BLK_DEV_OPTI621 is not set
CONFIG_BLK_DEV_RZ1000=y
CONFIG_BLK_DEV_IDEDMA_PCI=y
# CONFIG_BLK_DEV_IDEDMA_FORCED is not set
CONFIG_IDEDMA_PCI_AUTO=y
# CONFIG_IDEDMA_ONLYDISK is not set
CONFIG_BLK_DEV_ADMA=y
# CONFIG_BLK_DEV_AEC62XX is not set
# CONFIG_BLK_DEV_ALI15X3 is not set
# CONFIG_BLK_DEV_AMD74XX is not set
# CONFIG_BLK_DEV_ATIIXP is not set
# CONFIG_BLK_DEV_CMD64X is not set
# CONFIG_BLK_DEV_TRIFLEX is not set
# CONFIG_BLK_DEV_CY82C693 is not set
# CONFIG_BLK_DEV_CS5520 is not set
# CONFIG_BLK_DEV_CS5530 is not set
# CONFIG_BLK_DEV_HPT34X is not set
# CONFIG_BLK_DEV_HPT366 is not set
# CONFIG_BLK_DEV_SC1200 is not set
CONFIG_BLK_DEV_PIIX=y
# CONFIG_BLK_DEV_NS87415 is not set
# CONFIG_BLK_DEV_PDC202XX_OLD is not set
# CONFIG_BLK_DEV_PDC202XX_NEW is not set
# CONFIG_BLK_DEV_SVWKS is not set
# CONFIG_BLK_DEV_SIIMAGE is not set
# CONFIG_BLK_DEV_SIS5513 is not set
# CONFIG_BLK_DEV_SLC90E66 is not set
# CONFIG_BLK_DEV_TRM290 is not set
# CONFIG_BLK_DEV_VIA82CXXX is not set
# CONFIG_IDE_ARM is not set
# CONFIG_IDE_CHIPSETS is not set
CONFIG_BLK_DEV_IDEDMA=y
```

```
# CONFIG_IDEDMA_IVB is not set
CONFIG_IDEDMA_AUTO=y
# CONFIG_BLK_DEV_HD is not set
```

这里的选项太多，对我有用的只有那几个，所以可以省了不少的 bzImage 的空间

```
# SCSI device support
#
CONFIG_SCSI=m
# CONFIG_SCSI_PROC_FS is not set

#
# SCSI support type (disk, tape, CD-ROM)
#
CONFIG_BLK_DEV_SD=m
# CONFIG_CHR_DEV_ST is not set
# CONFIG_CHR_DEV_OSST is not set
# CONFIG_BLK_DEV_SR is not set
# CONFIG_CHR_DEV_SG is not set

#
# Some SCSI devices (e.g. CD jukebox) support multiple LUNs
#
# CONFIG_SCSI_MULTI_LUN is not set
# CONFIG_SCSI_REPORT_LUNS is not set
# CONFIG_SCSI_CONSTANTS is not set
# CONFIG_SCSI_LOGGING is not set

#
# SCSI Transport Attributes
#
# CONFIG_SCSI_SPI_ATTRS is not set
# CONFIG_SCSI_FC_ATTRS is not set

#
# SCSI low-level drivers
#
# CONFIG_BLK_DEV_3W_XXXX_RAID is not set
# CONFIG_SCSI_7000FASST is not set
# CONFIG_SCSI_ACARD is not set
# CONFIG_SCSI_AHA152X is not set
# CONFIG_SCSI_AHA1542 is not set
```

```
# CONFIG_SCSI_AACRAID is not set
# CONFIG_SCSI_AIC7XXX is not set
# CONFIG_SCSI_AIC7XXX_OLD is not set
# CONFIG_SCSI_AIC79XX is not set
# CONFIG_SCSI_ADVANSYS is not set
# CONFIG_SCSI_IN2000 is not set
# CONFIG_SCSI_MEGARAID is not set
# CONFIG_SCSI_SATA is not set
# CONFIG_SCSI_BUSLOGIC is not set
# CONFIG_SCSI_CPQFCTS is not set
# CONFIG_SCSI_DMX3191D is not set
# CONFIG_SCSI_DTC3280 is not set
# CONFIG_SCSI_EATA is not set
# CONFIG_SCSI_EATA_PIO is not set
# CONFIG_SCSI_FUTURE_DOMAIN is not set
# CONFIG_SCSI_GDTH is not set
# CONFIG_SCSI_GENERIC_NCR5380 is not set
# CONFIG_SCSI_GENERIC_NCR5380_MMIO is not set
# CONFIG_SCSI_IPS is not set
# CONFIG_SCSI_INIA100 is not set
# CONFIG_SCSI_NCR53C406A is not set
# CONFIG_SCSI_SYM53C8XX_2 is not set
# CONFIG_SCSI_IPR is not set
# CONFIG_SCSI_PAS16 is not set
# CONFIG_SCSI_PSI240I is not set
# CONFIG_SCSI_QLOGIC_FAS is not set
# CONFIG_SCSI_QLOGIC_ISP is not set
# CONFIG_SCSI_QLOGIC_FC is not set
# CONFIG_SCSI_QLOGIC_1280 is not set
CONFIG_SCSI_QLA2XXX=m
# CONFIG_SCSI_QLA21XX is not set
# CONFIG_SCSI_QLA22XX is not set
# CONFIG_SCSI_QLA2300 is not set
# CONFIG_SCSI_QLA2322 is not set
# CONFIG_SCSI_QLA6312 is not set
# CONFIG_SCSI_QLA6322 is not set
# CONFIG_SCSI_SYM53C416 is not set
# CONFIG_SCSI_DC395x is not set
# CONFIG_SCSI_DC390T is not set
# CONFIG_SCSI_T128 is not set
# CONFIG_SCSI_U14_34F is not set
# CONFIG_SCSI_ULTRASTOR is not set
```

CONFIG_SCSI_NSP32 is not set
CONFIG_SCSI_DEBUG is not set

如果你要使用移动硬盘，CONFIG_SCSI=m 是一定要选择的，而最好是将它做为一个 modules 来处理。因为我没别的 SCSI 设备了，所以几乎是全部 NO
还有一个漏的：CONFIG_SCSI_QLA2XXX=m，回家再编辑一下，去掉，这是个没用的东西，当时没注意

#
Old CD-ROM drivers (not SCSI, not IDE)
#
CONFIG_CD_NO_IDESCSI is not set

#
Multi-device support (RAID and LVM)
#
CONFIG_MD is not set

#
Fusion MPT device support
#
CONFIG_FUSION is not set

这些东西是做什么用的？我的 NOTEBOOK 用不到，全部 NO

IEEE 1394 (FireWire) support
#
CONFIG_IEEE1394=m

#
Subsystem Options
#
CONFIG_IEEE1394_VERBOSEDEBUG is not set
CONFIG_IEEE1394_OUI_DB is not set
CONFIG_IEEE1394_EXTRA_CONFIG_ROMS is not set

#
Device Drivers
#

#

```
# Texas Instruments PCILynx requires I2C
#
CONFIG_IEEE1394_OHCI1394=m

#
# Protocol Drivers
#
# CONFIG_IEEE1394_VIDEO1394 is not set
# CONFIG_IEEE1394_SBP2 is not set
# CONFIG_IEEE1394_ETH1394 is not set
CONFIG_IEEE1394_DV1394=m
CONFIG_IEEE1394_RAWIO=m
# CONFIG_IEEE1394_CMP is not set
```

因为我的电脑上有这个功能，还是编辑进来吧，如果有朋友也有这个协议，试一下这样选择看启动正常不？

```
#
# I2O device support
#
# CONFIG_I2O is not set
```

这个东东就不用了吧

```
# Networking support
#
CONFIG_NET=y

#
# Networking options
#
CONFIG_PACKET=y
CONFIG_PACKET_MMAP=y
# CONFIG_NETLINK_DEV is not set
CONFIG_UNIX=y
# CONFIG_NET_KEY is not set
CONFIG_INET=y
# CONFIG_IP_MULTICAST is not set
# CONFIG_IP_ADVANCED_ROUTER is not set
# CONFIG_IP_PNP is not set
# CONFIG_NET_IPIP is not set
# CONFIG_NET_IPGRE is not set
```

```
# CONFIG_ARPD is not set
# CONFIG_SYN_COOKIES is not set
# CONFIG_INET_AH is not set
# CONFIG_INET_ESP is not set
# CONFIG_INET_IPCOMP is not set
# CONFIG_IPV6 is not set
# CONFIG_NETFILTER is not set

#
# SCTP Configuration (EXPERIMENTAL)
#
# CONFIG_IP_SCTP is not set
# CONFIG_ATM is not set
# CONFIG_BRIDGE is not set
# CONFIG_VLAN_8021Q is not set
# CONFIG_DECNET is not set
# CONFIG_LLC2 is not set
# CONFIG_IPX is not set
# CONFIG_ATALK is not set
# CONFIG_X25 is not set
# CONFIG_LAPB is not set
# CONFIG_NET_DIVERT is not set
# CONFIG_ECONET is not set
# CONFIG_WAN_ROUTER is not set
# CONFIG_NET_FASTROUTE is not set
# CONFIG_NET_HW_FLOWCONTROL is not set

#
# QoS and/or fair queueing
#
# CONFIG_NET_SCHED is not set

#
# Network testing
#
# CONFIG_NET_PKTGEN is not set
# CONFIG_NETPOLL is not set
# CONFIG_NET_POLL_CONTROLLER is not set
# CONFIG_HAMRADIO is not set
# CONFIG_IRDA is not set
# CONFIG_BT is not set
CONFIG_NETDEVICES=y
```

```
CONFIG_DUMMY=m
# CONFIG_BONDING is not set
# CONFIG_EQUALIZER is not set
# CONFIG_TUN is not set
# CONFIG_NET_SB1000 is not set

#
# ARCnet devices
#
# CONFIG_ARCNET is not set

#
# Ethernet (10 or 100Mbit)
#
CONFIG_NET_ETHERNET=y
CONFIG_MII=y
# CONFIG_HAPPYMEAL is not set
# CONFIG_SUNGEM is not set
# CONFIG_NET_VENDOR_3COM is not set
# CONFIG_LANCE is not set
# CONFIG_NET_VENDOR_SMC is not set
# CONFIG_NET_VENDOR_RACAL is not set

#
# Tulip family network device support
#
# CONFIG_NET_TULIP is not set
# CONFIG_AT1700 is not set
# CONFIG_DEPCA is not set
# CONFIG_HP100 is not set
# CONFIG_NET_ISA is not set
CONFIG_NET_PCI=y
# CONFIG_PCNET32 is not set
# CONFIG_AMD8111_ETH is not set
# CONFIG_ADAPTEC_STARFIRE is not set
# CONFIG_AC3200 is not set
# CONFIG_APRICOT is not set
CONFIG_B44=y
# CONFIG_FORCEDETH is not set
# CONFIG_CS89x0 is not set
# CONFIG_DGRS is not set
# CONFIG_EEPRO100 is not set
```

```
# CONFIG_E100 is not set
# CONFIG_FEALNX is not set
# CONFIG_NATSEMI is not set
# CONFIG_NE2K_PCI is not set
# CONFIG_8139CP is not set
# CONFIG_8139TOO is not set
# CONFIG_SIS900 is not set
# CONFIG_EPIC100 is not set
# CONFIG_SUNDANCE is not set
# CONFIG_TLAN is not set
# CONFIG_VIA_RHINE is not set
# CONFIG_NET_POCKET is not set

#
# Ethernet (1000 Mbit)
#
# CONFIG_ACENIC is not set
# CONFIG_DL2K is not set
# CONFIG_E1000 is not set
# CONFIG_NS83820 is not set
# CONFIG_HAMACHI is not set
# CONFIG_YELLOWFIN is not set
# CONFIG_R8169 is not set
# CONFIG_SK98LIN is not set
# CONFIG_TIGON3 is not set

#
# Ethernet (10000 Mbit)
#
# CONFIG_IXGB is not set
# CONFIG_S2IO is not set

#
# Token Ring devices
#
# CONFIG_TR is not set

#
# Wireless LAN (non-hamradio)
#
# CONFIG_NET_RADIO is not set
```

```
#
# Wan interfaces
#
# CONFIG_WAN is not set
# CONFIG_FDDI is not set
# CONFIG_HIPPI is not set
CONFIG_PPP=m
CONFIG_PPP_MULTILINK=y
CONFIG_PPP_FILTER=y
CONFIG_PPP_ASYNC=m
CONFIG_PPP_SYNC_TTY=m
CONFIG_PPP_DEFLATE=m
CONFIG_PPP_BSDCOMP=m
CONFIG_PPPOE=m
# CONFIG_SLIP is not set
# CONFIG_NET_FC is not set
# CONFIG_SHAPER is not set
# CONFIG_NETCONSOLE is not set
```

这里有点长了，谁让它是一体呢？其实这里对我来说有用的就几项：
Ethernet (10 or 100Mbit)里面的 CONFIG_B44=y，这是我的网卡，现在可以大胆的把它编辑到内核里去了，如果在前面的：Code maturity level options 你没有选择的话，就可能找不到你的网卡哟。因我的电脑不是服务器，所以什么 IPV6 等这些协议全都没要，为什么？你看过有人用笔记本电脑做服务器的吗？
另外有一点要在这里提醒大家，如果你要用 ADSL 上网，这里你是必定要加进来的：

```
CONFIG_PPP=m
CONFIG_PPP_MULTILINK=y
CONFIG_PPP_FILTER=y
CONFIG_PPP_ASYNC=m
CONFIG_PPP_SYNC_TTY=m
CONFIG_PPP_DEFLATE=m
CONFIG_PPP_BSDCOMP=m
CONFIG_PPPOE=m
# CONFIG_SLIP is not set
# CONFIG_NET_FC is not set
# CONFIG_SHAPER is not set
# CONFIG_NETCONSOLE is not set
```

如果不加进来你试试能通过 ADSL 拨号上网才怪，这里说的是你直接使用 ADSL 拨号的情况，如果你用 LAN，算我没说

```
#
# ISDN subsystem
#
# CONFIG_ISDN is not set

#
# Telephony Support
#
# CONFIG_PHONE is not set

#
# Input device support
#
CONFIG_INPUT=y

#
# Userland interfaces
#
CONFIG_INPUT_MOUSEDEV=y
CONFIG_INPUT_MOUSEDEV_PSAUX=y
CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768
# CONFIG_INPUT_JOYDEV is not set
# CONFIG_INPUT_TSDEV is not set
# CONFIG_INPUT_EVDEV is not set
# CONFIG_INPUT_EVBUG is not set

#
# Input I/O drivers
#
# CONFIG_GAMEPORT is not set
CONFIG_SOUND_GAMEPORT=y
CONFIG_SERIO=y
CONFIG_SERIO_I8042=y
# CONFIG_SERIO_SERPORT is not set
# CONFIG_SERIO_CT82C710 is not set
# CONFIG_SERIO_PCIPS2 is not set

#
# Input Device Drivers
#
CONFIG_INPUT_KEYBOARD=y
```

```
CONFIG_KEYBOARD_ATKBD=y
# CONFIG_KEYBOARD_SUNKBD is not set
# CONFIG_KEYBOARD_LKKBD is not set
# CONFIG_KEYBOARD_XTKBD is not set
# CONFIG_KEYBOARD_NEWTON is not set
CONFIG_INPUT_MOUSE=y
CONFIG_MOUSE_PS2=y
# CONFIG_MOUSE_SERIAL is not set
# CONFIG_MOUSE_INPORT is not set
# CONFIG_MOUSE_LOGIBM is not set
# CONFIG_MOUSE_PC110PAD is not set
# CONFIG_MOUSE_VSXXXAA is not set
# CONFIG_INPUT_JOYSTICK is not set
# CONFIG_INPUT_TOUCHSCREEN is not set
# CONFIG_INPUT_MISC is not set
```

现在还有谁有使用 ISDN 上网或办公？
Telephony Support 这里的协议不是说要你选择来支持电话上网的，是支持网络电话的功能(是不是这样说的？)，现在 IP 电话这样便宜，何苦让自己的电脑受这份罪哟
在输入设置选择上，基本使用默认就可以了，但在
CONFIG_MOUSE_SERIAL is not set 上，原来是 YES 的，

```
# Character devices
#
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_HW_CONSOLE=y
# CONFIG_SERIAL_NONSTANDARD is not set

#
# Serial drivers
#
# CONFIG_SERIAL_8250 is not set

#
# Non-8250 serial port support
#
CONFIG_UNIX98_PTYS=y
# CONFIG_LEGACY_PTYS is not set
# CONFIG_QIC02_TAPE is not set

#
```

```
# IPMI
#
# CONFIG_IPMI_HANDLER is not set
```

说实话，这里我也是不太明白要怎么选择，好在这样的设置还能正常使用

```
#
# Watchdog Cards
#
# CONFIG_WATCHDOG is not set
# CONFIG_HW_RANDOM is not set
# CONFIG_NVRAM is not set
CONFIG_RTC=y
# CONFIG_DTLK is not set
# CONFIG_R3964 is not set
# CONFIG_APPLICOM is not set
# CONFIG_SONYPI is not set

#
# Ftape, the floppy tape device driver
#
# CONFIG_FTAPE is not set
CONFIG_AGP=y
CONFIG_AGP_ALI=y
# CONFIG_AGP_ATI is not set
# CONFIG_AGP_AMD is not set
# CONFIG_AGP_AMD64 is not set
CONFIG_AGP_INTEL=y
CONFIG_AGP_INTEL_MCH=y
# CONFIG_AGP_NVIDIA is not set
CONFIG_AGP_SIS=y
CONFIG_AGP_SWORKS=y
CONFIG_AGP_VIA=y
CONFIG_AGP_EFFICEON=y
CONFIG_DRM=y
CONFIG_DRM_TDFX=m
# CONFIG_DRM_GAMMA is not set
# CONFIG_DRM_R128 is not set
CONFIG_DRM_RADEON=m
CONFIG_DRM_I810=m
# CONFIG_DRM_I830 is not set
# CONFIG_DRM_MGA is not set
```

```
# CONFIG_DRM_SIS is not set
# CONFIG_MWAVE is not set
# CONFIG_RAW_DRIVER is not set
# CONFIG_HANGCHECK_TIMER is not set
```

在这里我只提醒一个地方：CONFIG_RTC=y，我不记得默认下是不是 YES 了，如果不选择启动的时候会有错误哟，干什么用的，忘了，看说明吧

```
#
# I2C support
#
# CONFIG_I2C is not set

#
# Misc devices
#
# CONFIG_IBM_ASM is not set

#
# Multimedia devices
#
# CONFIG_VIDEO_DEV is not set

#
# Digital Video Broadcasting Devices
#
# CONFIG_DVB is not set

#
# Graphics support
#
# CONFIG_FB is not set
# CONFIG_VIDEO_SELECT is not set

#
# Console display driver support
#
CONFIG_VGA_CONSOLE=y
# CONFIG_MDA_CONSOLE is not set
CONFIG_DUMMY_CONSOLE=y
```

I2C support、Misc devices、Multimedia devices 我都不用，所以是 NO，最后一个 Multimedia devices 朋友们要好好看看了，对使用 MP3 朋友可能有帮助哟，其它的基本默认

```
Sound
#
CONFIG_SOUND=y

#
# Advanced Linux Sound Architecture
#
# CONFIG_SND is not set

#
# Open Sound System
#
CONFIG_SOUND_PRIME=y
# CONFIG_SOUND_BT878 is not set
# CONFIG_SOUND_CMPCI is not set
# CONFIG_SOUND_EMU10K1 is not set
# CONFIG_SOUND_FUSION is not set
# CONFIG_SOUND_CS4281 is not set
# CONFIG_SOUND_ES1370 is not set
CONFIG_SOUND_ES1371=y
# CONFIG_SOUND_ESSSOLO1 is not set
# CONFIG_SOUND_MAESTRO is not set
# CONFIG_SOUND_MAESTRO3 is not set
CONFIG_SOUND_ICH=y
# CONFIG_SOUND_SONICVIBES is not set
# CONFIG_SOUND_TRIDENT is not set
# CONFIG_SOUND_MSNDCLAS is not set
# CONFIG_SOUND_MSNDPIN is not set
# CONFIG_SOUND_VIA82CXXX is not set
CONFIG_SOUND_OSS=m
CONFIG_SOUND_TRACEINIT=y
CONFIG_SOUND_DMAP=y
# CONFIG_SOUND_AD1816 is not set
# CONFIG_SOUND_AD1889 is not set
CONFIG_SOUND_SGALAXY=m
CONFIG_SOUND_ADLIB=m
CONFIG_SOUND_ACI_MIXER=m
CONFIG_SOUND_CS4232=m
CONFIG_SOUND_SSCAPE=m
```

```
CONFIG_SOUND_GUS=m
CONFIG_SOUND_GUS16=y
CONFIG_SOUND_GUSMAX=y
CONFIG_SOUND_VMIDI=m
# CONFIG_SOUND_TRIX is not set
CONFIG_SOUND_MSS=m
# CONFIG_SOUND_MPU401 is not set
# CONFIG_SOUND_NM256 is not set
# CONFIG_SOUND_MAD16 is not set
# CONFIG_SOUND_PAS is not set
# CONFIG_SOUND_PSS is not set
# CONFIG_SOUND_SB is not set
# CONFIG_SOUND_AWE32_SYNTH is not set
# CONFIG_SOUND_WAVEFRONT is not set
# CONFIG_SOUND_MAUI is not set
# CONFIG_SOUND_YM3812 is not set
# CONFIG_SOUND_OPL3SA1 is not set
# CONFIG_SOUND_OPL3SA2 is not set
# CONFIG_SOUND_YMFPCI is not set
# CONFIG_SOUND_UART6850 is not set
# CONFIG_SOUND_AEDSP16 is not set
# CONFIG_SOUND_ALI5455 is not set
# CONFIG_SOUND_FORTE is not set
# CONFIG_SOUND_RME96XX is not set
# CONFIG_SOUND_AD1980 is not set
```

唉，虽然 2.6.6 有了 ALAS 的支持，我还是没有用，为什么？因为我只要是有声音就行了，我想我的 NOTEBOOK 再好也没有 5.1 的功能，所以还是用 OSS 吧。其实这里有几个地方是没用的如：

```
CONFIG_SOUND_TRACEINIT=y
CONFIG_SOUND_DMAP=y
# CONFIG_SOUND_AD1816 is not set
# CONFIG_SOUND_AD1889 is not set
CONFIG_SOUND_SGALAXY=m
CONFIG_SOUND_ADLIB=m
CONFIG_SOUND_ACI_MIXER=m
CONFIG_SOUND_CS4232=m
CONFIG_SOUND_SSCAPE=m
CONFIG_SOUND_GUS=m
CONFIG_SOUND_GUS16=y
CONFIG_SOUND_GUSMAX=y
CONFIG_SOUND_VMIDI=m
# CONFIG_SOUND_TRIX is not set
```

CONFIG_SOUND_MSS=m

就是这里的几个选项，在编辑的时候报错说：编辑了也没用，为啥？因为我没那声卡呀。

```
# USB support
#
CONFIG_USB=y
# CONFIG_USB_DEBUG is not set

#
# Miscellaneous USB options
#
CONFIG_USB_DEVICEFS=y
# CONFIG_USB_BANDWIDTH is not set
# CONFIG_USB_DYNAMIC_MINORS is not set

#
# USB Host Controller Drivers
#
CONFIG_USB_EHCI_HCD=m
# CONFIG_USB_EHCI_SPLIT_ISO is not set
# CONFIG_USB_EHCI_ROOT_HUB_TT is not set
CONFIG_USB_OHCI_HCD=m
CONFIG_USB_UHCI_HCD=m

#
# USB Device Class drivers
#
# CONFIG_USB_AUDIO is not set
# CONFIG_USB_BLUETOOTH_TTY is not set
# CONFIG_USB_MIDI is not set
# CONFIG_USB_ACM is not set
# CONFIG_USB_PRINTER is not set
CONFIG_USB_STORAGE=m
# CONFIG_USB_STORAGE_DEBUG is not set
# CONFIG_USB_STORAGE_DATAFAB is not set
# CONFIG_USB_STORAGE_FREECOM is not set
# CONFIG_USB_STORAGE_ISD200 is not set
# CONFIG_USB_STORAGE_DPCM is not set
# CONFIG_USB_STORAGE_HP8200e is not set
# CONFIG_USB_STORAGE_SDDR09 is not set
# CONFIG_USB_STORAGE_SDDR55 is not set
```

```
# CONFIG_USB_STORAGE_JUMPSHOT is not set

#
# USB Human Interface Devices (HID)
#
CONFIG_USB_HID=m
CONFIG_USB_HIDINPUT=y
# CONFIG_HID_FF is not set
CONFIG_USB_HIDDEV=y

#
# USB HID Boot Protocol drivers
#
CONFIG_USB_KBD=m
CONFIG_USB_MOUSE=m
# CONFIG_USB_AIPTEK is not set
# CONFIG_USB_WACOM is not set
# CONFIG_USB_KBTAB is not set
# CONFIG_USB_POWERMATE is not set
# CONFIG_USB_MTOUCH is not set
# CONFIG_USB_EGALAX is not set
# CONFIG_USB_XPAD is not set
# CONFIG_USB_ATI_REMOTE is not set

#
# USB Imaging devices
#
# CONFIG_USB_MDC800 is not set
# CONFIG_USB_MICROTEK is not set
# CONFIG_USB_HPUSBSCSI is not set

#
# USB Multimedia devices
#
# CONFIG_USB_DABUSB is not set

#
# Video4Linux support is needed for USB Multimedia device support
#

#
# USB Network adaptors
```

```
#
# CONFIG_USB_CATC is not set
# CONFIG_USB_KAWETH is not set
# CONFIG_USB_PEGASUS is not set
# CONFIG_USB_RTL8150 is not set
# CONFIG_USB_USBNET is not set

#
# USB port drivers
#

#
# USB Serial Converter support
#
# CONFIG_USB_SERIAL is not set

#
# USB Miscellaneous drivers
#
# CONFIG_USB_EMI62 is not set
# CONFIG_USB_EMI26 is not set
# CONFIG_USB_TIGL is not set
# CONFIG_USB_AUERSWALD is not set
# CONFIG_USB_RIO500 is not set
# CONFIG_USB_LEGOTOWER is not set
# CONFIG_USB_LCD is not set
# CONFIG_USB_LED is not set
# CONFIG_USB_CYTHERM is not set
# CONFIG_USB_PHIDGETSERVO is not set
# CONFIG_USB_TEST is not set

#
# USB Gadget Support
#
# CONFIG_USB_GADGET is not set
```

KAO，现在 USB 的设备越来越多了，多的让你都找不到自己是哪个东西了，但最重要的是你要是想用 USB 的设备有几个地方是一定要选择的：

```
USB Host Controller Drivers
#
CONFIG_USB_EHCI_HCD=m
```

```
# CONFIG_USB_EHCI_SPLIT_ISO is not set
# CONFIG_USB_EHCI_ROOT_HUB_TT is not set
CONFIG_USB_OHCI_HCD=m
CONFIG_USB_UHCI_HCD=m
```

这里我劝大家不要 YES，为什么？不知道，不信你试试。

1、移动硬盘：CONFIG_USB_STORAGE=m，这个是不能少的哟
2、键盘、MOUSE：

```
# USB Human Interface Devices (HID)
#
CONFIG_USB_HID=m
CONFIG_USB_HIDINPUT=y
# CONFIG_HID_FF is not set
CONFIG_USB_HIDDEV=y

#
# USB HID Boot Protocol drivers
#
CONFIG_USB_KBD=m
CONFIG_USB_MOUSE=m
```

这几个项在以前我升级内核时不是这样选择的，启动时会有错误。
现在什么 DC ，移动硬盘呀，都可以使用了，不过有一个问题是我的 80G 移动硬盘找不到，只支持我的 20G 小硬盘，再试试。

```
# File systems
#
CONFIG_EXT2_FS=y
# CONFIG_EXT2_FS_XATTR is not set
CONFIG_EXT3_FS=y
# CONFIG_EXT3_FS_XATTR is not set
CONFIG_JBD=y
# CONFIG_JBD_DEBUG is not set
# CONFIG_REISERFS_FS is not set
# CONFIG_JFS_FS is not set
# CONFIG_XFS_FS is not set
# CONFIG_MINIX_FS is not set
# CONFIG_ROMFS_FS is not set
# CONFIG_QUOTA is not set
CONFIG_AUTOFS_FS=y
CONFIG_AUTOFS4_FS=y
```

```
#
# CD-ROM/DVD Filesystems
#
CONFIG_ISO9660_FS=y
# CONFIG_JOLIET is not set
# CONFIG_ZISOFS is not set
# CONFIG_UDF_FS is not set

#
# DOS/FAT/NT Filesystems
#
CONFIG_FAT_FS=y
CONFIG_MSDOS_FS=y
CONFIG_VFAT_FS=y
CONFIG_NTFS_FS=y
# CONFIG_NTFS_DEBUG is not set
CONFIG_NTFS_RW=y

#
# Pseudo filesystems
#
CONFIG_PROC_FS=y
CONFIG_PROC_KCORE=y
CONFIG_SYSFS=y
# CONFIG_DEVFS_FS is not set
CONFIG_DEVPTS_FS_XATTR=y
CONFIG_DEVPTS_FS_SECURITY=y
CONFIG_TMPFS=y
CONFIG_HUGETLBFS=y
CONFIG_HUGETLB_PAGE=y
CONFIG_RAMFS=y

#
# Miscellaneous filesystems
#
# CONFIG_ADFS_FS is not set
# CONFIG_AFFS_FS is not set
# CONFIG_HFS_FS is not set
# CONFIG_HFSPLUS_FS is not set
# CONFIG_BEFS_FS is not set
# CONFIG_BFS_FS is not set
```

```
# CONFIG_EFS_FS is not set
# CONFIG_CRAMFS is not set
# CONFIG_VXFS_FS is not set
# CONFIG_HPFS_FS is not set
# CONFIG_QNX4FS_FS is not set
# CONFIG_SYSV_FS is not set
# CONFIG_UFS_FS is not set

# Network File Systems
#
CONFIG_NFS_FS=y
# CONFIG_NFS_V3 is not set
# CONFIG_NFS_V4 is not set
# CONFIG_NFS_DIRECTIO is not set
CONFIG_NFSD=y
# CONFIG_NFSD_V3 is not set
# CONFIG_NFSD_TCP is not set
CONFIG_LOCKD=y
CONFIG_EXPORTFS=y
CONFIG_SUNRPC=y
# CONFIG_RPCSEC_GSS_KRB5 is not set
CONFIG_SMB_FS=y
CONFIG_SMB_NLS_DEFAULT=y
CONFIG_SMB_NLS_REMOTE="cp437"
# CONFIG_CIFS is not set
# CONFIG_NCP_FS is not set
# CONFIG_CODA_FS is not set
# CONFIG_AFS_FS is not set
```

文件系统这里我的选择就少了，其实默认的也差不多了，不过简单实用是最好的了，不多说，看说明吧

```
# Partition Types
#
# CONFIG_PARTITION_ADVANCED is not set
CONFIG_MSDOS_PARTITION=y

#
# Native Language Support
#
CONFIG_NLS=y
CONFIG_NLS_DEFAULT="utf8"
```

CONFIG_NLS_CODEPAGE_437=m
CONFIG_NLS_CODEPAGE_737=m
CONFIG_NLS_CODEPAGE_775=m
CONFIG_NLS_CODEPAGE_850=m
CONFIG_NLS_CODEPAGE_852=m
CONFIG_NLS_CODEPAGE_855=m
CONFIG_NLS_CODEPAGE_857=m
CONFIG_NLS_CODEPAGE_860=m
CONFIG_NLS_CODEPAGE_861=m
CONFIG_NLS_CODEPAGE_862=m
CONFIG_NLS_CODEPAGE_863=m
CONFIG_NLS_CODEPAGE_864=m
CONFIG_NLS_CODEPAGE_865=m
CONFIG_NLS_CODEPAGE_866=m
CONFIG_NLS_CODEPAGE_869=m
CONFIG_NLS_CODEPAGE_936=y
CONFIG_NLS_CODEPAGE_950=m
CONFIG_NLS_CODEPAGE_932=m
CONFIG_NLS_CODEPAGE_949=m
CONFIG_NLS_CODEPAGE_874=m
CONFIG_NLS_ISO8859_8=m
CONFIG_NLS_CODEPAGE_1250=m
CONFIG_NLS_CODEPAGE_1251=m
CONFIG_NLS_ISO8859_1=m
CONFIG_NLS_ISO8859_2=m
CONFIG_NLS_ISO8859_3=m
CONFIG_NLS_ISO8859_4=m
CONFIG_NLS_ISO8859_5=m
CONFIG_NLS_ISO8859_6=m
CONFIG_NLS_ISO8859_7=m
CONFIG_NLS_ISO8859_9=m
CONFIG_NLS_ISO8859_13=m
CONFIG_NLS_ISO8859_14=m
CONFIG_NLS_ISO8859_15=m
CONFIG_NLS_KOI8_R=m
CONFIG_NLS_KOI8_U=m
CONFIG_NLS_UTF8=m

咋还是这个东东？
Partition Types
#
CONFIG_PARTITION_ADVANCED is not set

CONFIG_MSDOS_PARTITION=y

语言支持建议全选择吧，当然中文你可以直接编进内核了，加载烦呀。

Profiling support
#
CONFIG_PROFILING is not set

#
Kernel hacking
#
# CONFIG_DEBUG_KERNEL is not set
CONFIG_EARLY_PRINTK=y
# CONFIG_DEBUG_SPINLOCK_SLEEP is not set
# CONFIG_FRAME_POINTER is not set
# CONFIG_4KSTACKS is not set

#
Security options
#
CONFIG_SECURITY is not set

#
Cryptographic options
#
CONFIG_CRYPTO is not set

#
Library routines
#
# CONFIG_CRC32 is not set
# CONFIG_LIBCRC32C is not set
CONFIG_ZLIB_INFLATE=m
CONFIG_ZLIB_DEFLATE=m
CONFIG_X86_BIOS_REBOOT=y
CONFIG_X86_STD_RESOURCES=y
CONFIG_PC=y

CONFIG_4KSTACKS is not set 这是大家都关心的一个选项了，还等什么，NO 呀。
至于说别的选项差不我都 NO 了，为啥？减少 bzImage 的体积呀

选择完了是不是？再看看吧，MD，我看了几次还是在 SCSI 那给多了一个没用的东西，好在不影响全局。
make bzImage
make modules
make modules_install
/sbin/depmod -a
make install(这一步也可以手动完成)

vi /etc/grub.conf
将
title Fedora
root (hd0,4)
kernel /vmlinuz-2.6.6 ro root=/dev/hda1
这里的 hda1 修改成你的 / 所在位置就可以了。
以 TEXT 模式启动，安装 NVIDIA 吧，不行的话你打电脑吧

71 ☆单独编译一个模块

把.config 文件拿来修改，只保留你需要的模块的内容，然后 make modules

如果你的 Kernel 配置支持 Modules 的话,解决那些问题是比较简单的
只要编译那些.o 文件就可以啦.

比如我的声卡经常 Irq 和 IO 不对, 我就到 /usr/src/linux/drivers/sound 目录下.

gcc -o configure configure.c
./configure

选好 IO,IRQ 等等,
make
cp sound.o /lib/modules/2.0.34/misc

rmmod sound
insmod sound init_trace=1
测试一下.
直到成功为止.

72 ☆内核补丁

内核补丁程序升级
zcat patch-x.gz|patch -p0

bzip2 -dc patch-2.6.6-bk5.bz2|patch -p1

内核补丁(以前的方法)
bzip2 -dc patch-xxx.bz2 | patch
PATCH 文件拷贝到/usr/src 下：
#patch -p0 < patch-2.2.16
#gzip -cd patch-2.4.x-pre2-ac1.gz|patch -p1 -s -N -E -d 源码目录
可以使用这种方式来安装任何补丁，而不用管它的文件名了

73 ☆sysctl

两种方法来配置一个已经存在的内核：sysctl 和 boot loader

sysctl 允许你查看内核所用的所有参数，甚至去设置那些参数(通常又叫 sysctls).sysctl 有着非常强大的作用，你可以使用它解决很多问题，而不用重新编译内核或者重新配置应用程序。但它有时也是应该让用户头痛的家伙

语法:
sysctl [-n] [-e] variable ...
sysctl [-n] [-e] [-q] -w variable=value ...
sysctl [-n] [-e] [-q] -p <filename>
sysctl [-n] [-e] -a
sysctl [-n] [-e] -A

参数:
variable
键名
例如:kernel.ostype
/符号可以替代.符号
如:kernel/ostype

variable=value
设置键值,用这种形式 variable=value,如果键值中包含引号或可以被 shell 解析的字符,就可以把键值放在双引号中,这种形式的赋值需要用-w 参数来使用.(等号前后不能有空格)

-n 只显示键值,不显示键名

-e 忽略未知键值的错误
-N 只显示键名,不显示键值
-q 不显示,就是不输出到标准输出
-w 改变 sysctl 设置
-p 如果没有指定文件就装入/etc/sysctl.conf 的设置,否则装入指定文件,改动文件后用这个装载
-a 显示当前所有可用的键
-A 用表格形式显示当前所有可用的键

例子
/sbin/sysctl -a
/sbin/sysctl -n kernel.hostname
/sbin/sysctl -w kernel.domainname="panda.com"
/sbin/sysctl -p /etc/sysctl.conf

所有的操作都有 sysctl 命令完成，但在进行操作前，你必须明白有那些可用的 sysctls 及其具体含义。你可以用下面的命令查看系统中所有可用的 sysctls
#sysctl -A >sysctl.out
这个命令运行后，sysctl.out 文件中就会出现许多 sysctls 及其参数，许多东西很难理解，但也有一些很容易看懂的。

kernel.hostname = panda
这个特殊的 sysctl 名叫 kern.hostname，其值为 panda 我运行此命令的机子叫 panda.

从 sysctl 的名字，可以很容易就猜到这是运行这台计算机上的内核的名字。真的那么容易理解吗？不！有一些却非常古怪：(这种参数已经逐渐取消了)
p1003_1b.memory_protection = 0
做为一个用户，我无法理解这个参数的意思。当然，如果我有问题并且想从软件商那或者邮件列表上获得帮助，我可以在请求帮助时顺便也把它附上。他们也许会告诉我去调整那个参数，以更好地支持该软件。所有的 sysctls 被组织在一个树（叫管理信息库 MIB--Management Information Base ）的格式中 (MIB 也用于系统管理的其他方面，我们会在本书中看到其他的例子.)。
这棵树由几个主要的目录，比如 net, vm, and kern. 每个目录又进一步细分，比如 net 包括所有的与网络有关的 sysctls,它又被细分为 IP, ICMP, TCP, 和 UDP.
这些 sysctls 描述了内核的 IPC 行为.这个 sysctl 的分支可以分几层继续下去 。

一些 sysctl MIB 数的根目录列与下表.

Some roots of the sysctl MIB tree
Sysctl | Functions

kern | 系统核心
vm | 虚拟内存
vfs | 文件系统

net | 网络
debug | 调试信息
hw |硬件信息
machdep |系统平台;依赖性变量(比如 Alpha,i386, 等)
user |用户的接口信息
p1003_1b | POSIX behavior

每个 sysctl 参数值可以是一个 string, integer, binary value,或者 opaque.
Strings 是自由格式的任意长度的文本;
integers 是一般的全数字;
binary values 是 0 (off) 或 1 (on);
opaques 在机器码中并且只有特殊才程序才能解释。

要想得到 sysctl 的某个确切的参数值，你需要将完整的 MIB 作为以上命令的参数，如：
# sysctl kernel.ostype
kernel.ostype = Linux

修改 Sysctls
一些 sysctl 的值是只读的。比如,看看 MIB 树 hw (hardware) 和 machdep(machine dependencies)。
kernel.ostype = Linux

此项控制用户是否可以 mount 诸如 CD-ROM 和 floppy 之类的介质，是可以被修改的。他默认被设置为 0 或者 off，为了打开它，用 sysctl 的-w 参数把它设为 1 即可。By default it is set to 0, or off. To turn it on, use use sysctl's ?w flag to set it to 1.
sysctl -w kernel.hostname=panda
Sysctl 返回一些信息，显示原来的值和修改后的值。

在启动时设定 Sysctl
想要在系统启动时即设定的 Sysctls 需要写入文件 /etc/sysctl.conf. 把你要设置的 sysctl 项及欲设值列在 sysctl.conf 文件中.

举个例子：
通过设置 kernel.hostname=panda 来让用户可以 mount 文件系统，在 sysctl.conf 文件中加入如下行：
kernel.hostname=panda

通过 Loader.conf 配置内核(BSD 中)

一些内核的配置必须发生在在系统启动前。比如当内核初始化检测 IDE 硬盘时，设备驱程决定要不要使用写缓存。这个决定必须在该设备首次侦测时。在系统启动时或者启动之后你都不可以改变你的决定！类似的，你可能有一块新的网卡并且想在系统启动前载入内核模块作为它的驱程. 这就是 system loader 的目的了。

loader 有很多作用: 它寻找硬件驱程包括内核，将内核载入到内存，启动 boot 进程、把信息传给内核。loader 给内核的信息中最重要的部分是在启动时即设置的 sysctl MIBs 。
配置系统 laoder 最普通的方法是编辑它的配置文件，虽然你可以在 loader 的 ok>提示符下手工输入命令载入内核 modules.但对于长的设置项来说，最好的方法还是将其写入/boot/loader.conf 中.FreeBSD 中有两个重要的 loader.conf 文件：/boot/loader.conf 和/boot/defaults/loader.conf.在这里我们只修改/boot/loader.conf.
在/boot/defaults/loader.conf 中的设置项都是系统的默认设置项，
在/boot/loader.conf 中的设置会覆盖系统的默认设置。

Loader.conf 有两个主要作用：载如内核模块和提供设备驱程的 hints。
Device driver 是普通的只在启动时设定的 sysctl MIBs .看看/boot/defaults/loader.conf,你会发现很多在各种环境下的有用的选项，比如可以指定一个非系统默认的内核 ，可以指定系统启动时的详细信息。作为参考，下面是/boot/defaults/loader.conf.的一点摘录：

```
kernel="/kernel"
kernel_options=""
userconfig_script_load="NO"
userconfig_script_name="/boot/kernel.conf"
userconfig_script_type="userconfig_script"
```

要改变启动的某项设置，你可以将其中相应行复制到/boot/loader.conf, 并在那修改它。
举个例子，上面的列表的第一项是设置内核名字的（这个我们在 sysctl 的例子中已经见过）。假如你工作在一台远程计算机上，你想在下次重起时使用不同的内核，但你又不想 cp 内核到/kernel. 你就可以修改该行来改变你的系统所使用的内核。这是一个只可以在 boot 时设置的 sysctl。

传递 hints 给设备驱程

Loader.conf 的首要目的是传递 hints 给设备驱程. (一个 hints 能否被某设备驱程利用,在 man page 中有说明)。
如前所述，IDE 硬盘的设备驱程在系统启动前就要决定是否使用写缓存
为了起用写缓存在 loader.conf 文件中写入：：

```
hw.ata.wc="1"
```

就可以了，这种类型的 flag 看起来就会怀疑是不是 sysctl MIB。事实上，系统启动时，首先看看它到底是不是一个 sysctl:

```
# sysctl hw.ata.wc
hw.ata.wc: 1
#
```

我们知道，它确实是!

当系统在运行时，你不可以改变 sysctl. (你可以试试，并无大碍）。你可以在启动时修改一个只读 sysctl. 但这个并不能让你把老的 Pentium 调整到一个 Alpha，它仅仅给你额外的灵活性 (flexibility)。

自动载入内核模块

如前所述，内核模块也是内核的一部分，在需要的时候可以启动（或者说载入），没用的时候又可以卸载。这样可以节省系统内存并且增强系统的灵活性。
在系统启动时自动载入内核模块是非常简单的。默认的 loader.conf 给出可很多例子。
将模块名 cp 到 loader.conf, 去掉后面的".ko", 并且加入类似的语句*_load="YES". 比如，要在系统启动时自动载入 /module/procfs.ko ， 只需在 loader.conf 中加入:
procfs_load="YES"

当然，最难的是知道该载入哪个模块。简单是是设备驱程，假如你添加了新的网卡或者 SCSI 卡，你就应该在 boot 系统启动时载入相应的内核模块，而不是重建内核。你可以通过程序的文档或者别人的建议来得知能不能通过载入 modules 来解决你的问题。怎么知道该载入哪个模块完全凭经验、看文档并且知道你到底想让你的系统做什么。

74 ☆sysctl.conf

/etc/sysctl.conf
调整内核

语法

注释
; 注释

键名 = 键值

空白行被忽略,键名或者键值前后的空格被忽略,尽管键值可以包含空格

例子
sysctl.conf sample
#

kernel.domainname = panda.com
; this one has a space which will be written to the sysctl!
kernel.modprobe = /sbin/mod probe

sysctl.conf 文件如下:
Kernel sysctl configuration file for Red Hat Linux
#
For binary values, 0 is disabled, 1 is enabled. See sysctl(8) and

```
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1
```

75 ☆Linux 单用户模式

单用户模式不要求我们输入 root 用户的密码，如果您丢失了 root 用户的密码，并能用单用户模式来重设您的 root 密码;

另外单用户模式还有一个前提是您的 grub 或者 lilo 是能正常工作的,并且您知道您的系统问题发生在哪里
当以单用户模式登录时，您能打开文件系统的写操作，然后进行您想要进行的系统修复;

有时候系统出现错误，系统管理员会采取的第一步措施就是启动至单用户模式。这个时候系统只是快速启动到一个 root shell 下而不必启动那些不必要的服务。

单用户模式下可以做的工作如下：
*系统崩溃后检查文件系统
*修改一些关键的配置文件，如/etc/fstab,/etc/passwd 以及/etc/inittab 等等
*通过先前的备份恢复系统

76 ☆GRUB 下单用户登录

这种方式进行引导就复杂一些，进入 GRUB 启动画面时按"C"进入 GRUB 命令行，有密码时按"P"之后输入密码之后再进行 GRUB 命令行。

在 grub 启动后，移动键盘到 Linux 的启动项;按 e 键;然后再移动键盘到类似下面的一行，也就是 kernel 的那行:

kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/1 rhgb quiet
把光标移动这行后，再按一下 e 键，进入编辑这行;在行尾条一个空格 ，然后输入 linux single，也就是类似如下的:
kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/1 rhgb quiet linux single
结束编辑，按回车返回;
接着我们要启动系统，按一下 b 键启动;
sh-3.00#
您可以用 df -h 来查看文件系统挂载位置等，也可以用 fdisk -l 来查看分区等;但对文件系统的写操作，可能还要看下面的;

单用户进入系统后，可能系统是只读的;要运行下面的命令;
#mount -o remount,rw /

kernel /boot/vmlinuz root=/dev/hda3 -s
也可以-s 参数启动到单用户模式

77 ☆LILO 下单用户登录

在 boot:处输入
linux single
您可以用 df -h 来查看文件系统挂载位置等，也可以用 fdisk -l 来查看分区等;但对文件系统的写操作，可能还要看下面的;

单用户进入系统后，可能系统是只读的;要运行下面的命令;
#mount -o remount,rw /

78 ☆软盘下单用户登录

在软盘启动之后出现"BOOT:"时，可以对启动的参数进行设置，在这里键入"linux single"之后启动系统即可以进行单用户方式。

79 ☆防止进入单用户

由于单用户对系统有完全的控制权限，如果操作不当或被他人进入，那么后果将不堪设想，如何防止入行单用户了，有以下几个注意的方面。
1、对/etc/inittab 文件进行保护。对/etc/inittab 文件，以 root 身份进入通过 chown 700 /etc/inittab 把属性设为其它用户不能修改就行了。
2、如果是使用的 lilo 方式进行引导，可能通过 linuxconf 或直接修改 lilo.conf 把引导时等待输入时间设置为 0 或最短时行。这种情况下，如果进入单用户方式，可以用软盘进行引导。
3、如果使用是 GRUB 方式进行引导，最简单的方法是使用 GRUB 密码，对启动选项进行保护。也可以设置等待时间,同上.

4、为了防止他人远程进行破坏，使系统重启，除了对 ROOT 的密码和/etc 目录下的文件进行有效管理之外，还应当对 CMOS 进行密码设置，这样即使把系统改成单用户方式了,也无法直接的启动计算机进行操作。

80 ☆Linux 修复模式

Linux 的修复模式就是 linux rescue;
修复模式并不需要系统密码，当进入修复模式后，系统会提示我们要挂载哪个分区上的文件系统，我们根据自己的系统情况来选择就就是了。登录完成后，系统也会提示所在分区的文件系统会被挂载到哪个目录;如果您不知道挂载在哪里，请通过 df -h 来查看;

修复模式需要安装盘的第一张或独立的修复盘;
比如 Fedora 就有专门的修复盘;其实修复盘在一定意义上来说类似 livecd，是不需要安装到硬盘也能运行系统;然后通过 mount 来挂载文件系统;

执行修复模式时，需要以下几个步骤;
其一:在 BIOS 中，设置首个启动驱动器为 CDROM;
其二;找出系统安装盘的第一张;放入 CDROM ;当光盘运行时，会提示安装的界面;
在 boot:后面输入 linux rescue ，然后按回车;
boot:linux rescue

GRUB 中进入 rescue 模式
kernel /boot/vmlinuz root=/dev/hda3 -b
内核参数-b 来进入救援模式。这个模式下并不像通常那样把文件系统挂载为读写模式，这就需要自己重新进行挂载。

进入修复模式，我们能做些什么呢？？什么都可以做，比如挂载 usb 盘进行数据备份;修改系统中的配置文件... ... 重新设置 root 密码、挂载文件系统 ... ... 只要能想到，大多都可以完成;

linux rescue 从本质上来说，就是跨越控制台登录，因为无需要密码验证就能登录系统;

linux rescue 下不能切换光盘
运行：umount /mnt/source
报错说，设备正在使用。----------因为/mnt/source 本来是没有的，通过 RESCUE 模式虚拟的一个 MNT！！！除非你退出 RESCUE 模式，
在 EXIT RESCUE 模式的时候有 UMOUNTING 的执行顺序

推荐你用软盘启动 RESCUE 模式(没试过)
你可以手动 mount 分区
当然也能换 cdrom 了
用“eject”这个命令可以弹出光盘!

81 ☆跨越控制台登录

真正的跨越控制台登录是不需要输入 root 密码的，如果您把 root 密码丢失，就要想到跨越控制台登录;我们前面所说的 linux rescue 从本质上来说，就是跨越控制台登录，因为无需要密码验证就能登录系统;

跨越控制台登录主要包括:借 grub 和 lilo 跨越控制台登录;linux rescue 模式;第三方 livecd 系统和第三方 Linux 系统;

82 ☆GRUB 跨越控制台登录

在 grub 启动后，移动键盘到 Linux 的启动项;按 e 键;然后再移动键盘到类似下面的一行，也就是 kernel 的那行:
kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/1 rhgb quiet
把光标移动这行后，再按一下 e 键，进入编辑这行;在行尾条一个空格 ，然后输入 linux single，也就是类似如下的:
kernel /boot/vmlinuz-2.6.11-1.1369_FC4 ro root=LABEL=/1 rhgb quiet linux init=/bin/bash
结束编辑，按回车返回;
接着我们要启动系统，按一下 b 键启动;
可能系统是只读的;要运行下面的命令;
#mount -o remount,rw /

也可以用(老版本):
出现 grub 菜单后
在/boot/vmlinuz
的后面加个“1”
如
/boot/vmlinuz-2.4.26-1-386 1 root=/dev/hda2 ro
这样也行

然后我们可以通过 df -h 来查看文件系统的加载情况;既然都把文件系统挂载了，有什么活干不了的呢？重设 root 密码，备份文件... ...

比如重设 root 密码;
#passwd

83 ☆LILO 跨越控制台登录

在 boot:处输入
boot:linux init=/bin/bash

当进入系统后，您可以用 df -h 来查看文件系统挂载位置等，也可以用 fdisk -l 来查看分区等;但对文件系统的写操作，可能还要看下面的;

也可以用(老版本):
启动出现 boot：的时候,键入
linux 1
就可以了

可能系统是只读的;要运行下面的命令;
#mount -o remount,rw /

84 ☆linux 的 livecd 光盘或第三方 linux 系统下跨越控制台登录

linux 的 livecd 版本是无需安装的 linux 系统，在光盘上就可以运行的 linux 系统;livecd 大多是用来修复之用，livecd 集成了常用的系统操作工具;
第三方 Linux 系统是指你的机器上安装一个以上的 Linux 系统，如果其中一个发生问题，我们就可以用另一个来修复;

livecd 和第三方 linux 系统来修复已经被破坏的 Linux，常用的工具有 mount 和 chroot 等等 ;如果您只是简单的更改存在问题的系统文件，用 mount 就足够了。chroot 工具可以改变/，进而进入另一个系统，这个工具的确有用。

通过 livecd linux 或其它 linux 来修复已经破坏的 Linux 步骤如下:

1)mount 挂载文件系统;
注:执行 mount 及 chroot 时，要以 root 权限运行，以下同;

比如存在问题的系统位于 /dev/hda5 ，如果您不知道哪个分区是 linux 的，您可以通过 fdsik -l 来查看;
[root@localhost ~]# fdisk -l 注:查看分区情况;
[root@localhost ~]# pwd 注:确定当前工作目录;
/root 注:当前工作目录为/root
[root@localhost ~]# mkdir systmp 注:在当前工作目录下创建一个临时目录;
[root@localhost ~]# mount /dev/hda5 systmp/ 注:挂载 hda5 到 tmpsys 目录中;
[root@localhost ~]# df -h 注:查看是否已经挂载;
Filesystem 容量 已用 可用 已用% 挂载点
/dev/hda8 11G 9.8G 581M 95% /
/dev/shm 236M 0 236M 0% /dev/shm
/dev/hda5 7.9G 5.9G 2.0G 76% /root/systmp 注:看好了，是挂载到了/root/systmp 目录中;

既然把/dev/hda5 分区的文件系统已经挂载了，我们就能对已经挂载的文件系统进行写操作;如果是简单的备份和文件修改，是完全能行的;

注意:在写操作的过程中，要倍加不心，在操作某个文件的时候，要先备份;当出现不能 mount 的情况，可能是您的当前所用的内核不支持相应的文件系统;

2)chroot 到已经挂载的文件系统;
chroot 这个工具很有用，很多 Linux 都支持 chroot 切换到另一个文件系统中，然后进行一系列系统包的安装和系统配置;这并不是说任何文件系统都能 chroot 切入 ，首先这个文件系统中得具备系统运行的一定的工具和环境，比如/bin 和/sbin 目录是拥有等....

比如我的机器中有两个 Linux 系统，一个运行正常，另一个有点问题(比如是 root 密码忘记了)，我想通过正常的系统来修复另一个;首先我们进行的是 mount ，也就是前面所说的，然后下一步就是 chroot ;通过 chroot 切换到要修复的文件系统中;

比如存在问题的系统位于 /dev/hda5
[root@localhost ~]# fdisk -l
[root@localhost ~]# pwd
/root
[root@localhost ~]# mkdir systmp
[root@localhost ~]# mount /dev/hda5 systmp/
[root@localhost ~]# df -h
Filesystem 容量 已用 可用 已用% 挂载点
/dev/hda8 11G 9.8G 581M 95% /
/dev/shm 236M 0 236M 0% /dev/shm
/dev/hda5 7.9G 5.9G 2.0G 76% /root/systmp
[root@localhost ~]# chroot systmp 注:chroot 到 hda5 分区的系统中;
bash-3.00# 注:已经登录;
bash-3.00# df -lh 注:查看文件系统挂载情况;
Filesystem 容量 已用 可用 已用% 挂载点
/dev/hda5 7.9G 5.9G 2.0G 76% /
proc 7.9G 5.9G 2.0G 76% /proc
sysfs 7.9G 5.9G 2.0G 76% /sys

上面我们就通过 chroot 命令完成了通过一个系统到另一个系统的跨控制台登录;Livecd 如果要完成此任务，也是通过这样过程完成的。

我们既然已经登录到有问题的系统了，可以进行相应的修复工作，比如对 root 密码的恢复，软件包的安装，相应文件的修改... ...

有些发行版的安装盘类似 livecd ，比如 slackware 的安装盘的第一张，当他启动到让你输入用户名和密码进行安装时，我们不必输入什么，或者直接输入 root ，就能进入 cdrom 虚拟环境了;这时我们就用前面所说 mount 加载文件系统，然后 chroot 挂载的文件系统;

85 ☆chroot

chroot 工具是 Linux 操作系统都具备的工具，从表面的意思看，chroot 是从一个/根到另一个/根。在一个 Linux 操作系统中安装另一个操作系统，就是利用 chroot 的这个特点。首先创建 chroot 运行的基础环境，然后通过 chroot 到新的/根，然后再用相应的软件包管理工具把新的操作系统其它软件包安装上;

几乎所有的开源操作系统都可以用 chroot 的方法来安装，比如 Gentoo 、LFS 及 CRUX 比较常用 chroot;通过 chroot 安装操作系统，好象堆积木一样，一块一块的把操作系统安装起来，所以 Gentoo 和 LFS 的 Fans 特别有成就感。呵，也说是操作系统全手工打造，成就感自然不言而喻了;

86 ☆chroot 安装操作系统

chroot 方式安装操作系统所需要的系统环境:
得有一个能运行的 Linux 操作系统或一个 LiveCD;

安装目的地的介质分为两种，一种是物理硬盘;一种是映像文件。
安装在物理硬盘的可以通过 GRUB 或 LILO 等引导管理器来独立真实运行
安装在映像文件中的能通过 chroot 访问或 Xen 虚拟运行;(就像 VMWARE 中把操作系统装在一个文件上)
安装在映像文件中的操作系统，可以用来学习，通过 chroot 访问，也可以通过 xen 来虚拟等 ... ... 还可以做成类似 file.iso 的形式来存储文件等;

通过 chroot 安装 linux 到一个硬盘物理分区中;

比如我的硬盘/dev/hda5 是空白分区，并且我想创建/dev/hda5 为 ext3 文件系统，并且把它挂载到 /mnt/slack 目录中，然后通过 chroot 来安装 Slackware 10.2;
第一步：您要通过分区工具 fdisk 或 parted 来规划您的分区，此步省略;
第二步：创建文件系统;
[root@localhost ~]#mkdir /mnt/slack
[root@localhost ~]# mkfs.ext3 /dev/hda5 注：格式化/dev/hda5 为 ext3 文件系统;
第三步：挂载文件系统;
[root@localhost ~]# mount /dev/hda5 /mnt/slack 注：挂载/dev/hda5 到 /mnt/slack 目录;
第四步：挂载 slackware 10.2 光盘的第一张，安装基础系统;
[root@localhost ~]# mount -o loop slackware-10.2-install-d1.iso /mnt/cdrom/
我们可以先在/mnt/slack 目录中创建一个软件包存放目录，用于存放 slackware-10.2-install-d1.iso(是个 LiveCD)中的所有内容;
[root@localhost ~]#mkdir /mnt/slack/pack
[root@localhost ~]# cp -rp /mnt/cdrom/* /mnt/slack/pack
然后把 slackware 第一张盘中的 slackware 目录中的 a 目录中的所有包都进行解压缩，以及 d 目录中 glibc 开头的包，d 目录中的 zlib 以及 zsh 并且把解出来目录，比如 /usr、etc、lib 等所有目录都复制到 /mnt/slack 中;

第五步：解决依赖关系;
如果我们运行下面的命令提示没有/bin/bash 这个文件时，我们要解决依赖关系;通过解决依赖关系，也会发现运行 chroot 所需要的必备的基础软件包;
[root@localhost ~]# chroot /mnt/slack
如果提示缺少一些文件，我们可以自行判断缺少哪些包;当我们进入/mnt/slack/bin 目录发现，的确没有 bash 这个文件，所以我们要复制一个 bash 过去;
[root@localhost ~]# cd /mnt/slack/bin
[root@localhost bin]# cp bash2.new bash
这时我们还要判断 bash 所依赖的库文件;
[root@localhost bin]# ldd bash
出来的依赖关系，大多是 glibc 的，glibc 被安装在了/mnt/slack/lib/tls 目录中，我们可以根据提示一个一个的做链接。少什么东西就做什么的链接;要看/mnt/slack/lib 中是否有 glibc 的文件，然后做链接。如果是存放在 tls 目录中的，也要链到/mnt/slack/lib 中。链接时要用相对路径，不能用绝对路径;
创建链接文件和在 Windows 创建快捷方式比较相似，用 ln 命令;
#ln -s 原文件名 新文件名
什么才算解决了 bash 的依赖关系了呢？直到能 chroot /mmt/slack 才算成功;
第六步： chroot 成功，进入 Slackware 系统;
chroot 成功后，我们就能进入 Slackware 系统，然后通过 pkgtool 或 installpkg 工具来安装其它的软件包，比如 内核什么的;
[root@localhost ~]# chroot /mnt/slack
第七步;更改/etc/fstab 文件;
如果要让一个操作系统独立运行，非得写一写/etc/fstab 文件，对于这个您可以参考，比如 下面的这个例子，您改一改就可以用了;
/dev/hda7 swap swap defaults 0 0
/dev/hda6 / reiserfs defaults 1 1
/dev/hda1 /mnt/winc ntfs ro 1 0
/dev/hda3 /mnt/wind vfat defaults 1 0
/dev/cdrom /mnt/cdrom auto noauto,owner,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
proc /proc proc defaults 0 0
none /sys sysfs defaults 0 0
第八步：引导系统;
要通过 GRUB 或 LILO 引导管理器实现对 Slackware 的引导;

通过 chroot 把 Linux 安装在一个映像文件中;
第一步：创建映像文件;
比如 我们创建一个名为 slack.img 的映像文件，体积为 2G 的，就可以用下面的命令;bs 是每个块的大小为 1M，共创建 2000 块;
[root@localhost ~]# dd if=/dev/zero of=slack.img bs=1M count=2000 seek=1024
第二步：创建文件系统;

我们可以创建为 ext3、fat32 或 reiserfs 等文件系统，创建文件系统
[root@localhost ~]# /sbin/mkfs.ext3 slack.img
mke2fs 1.38 (30-Jun-2005)
slack.img is not a block special device.
Proceed anyway? (y,n) y
第三步：挂载已被格式化映像文件;
[root@localhost ~]# mkdir /mnt/slack
[root@localhost ~]# mount -o loop slack.img /mnt/slack/
第四步以后和物理硬盘操作基本相同，省略过去;

chroot 成功切入新的操作系统后的软件安装
如果通过 chroot 能进入新安装的操作系统，我们就可以新操作系统的软件包管理工具来安装其它软件包了，比如 Slackware 用的是 pkgtool 或 installpkg 工具;Fedora 用的是 rpm 等工具;大多发行版所用的工具不尽相同，以发行版为准 ；

有关软件包 rpm、tgz、deb 等软件包提取;
发行版都有一定的软件包格式，比如 file.rpm 、file.deb 或 file.tgz 或 file.tar.gz 等;rpm 格式的软件包，一般基于 Redhat 或 Fedora 为基础开发的都采用 RPM 格式。因为 chroot 安装操作系统，首要的是从一个软件包中提取文件，然后复制到文件系统中，所以提取文件工具也得做一点解说;

从 rpm 软件包抽取文件;
操作的前提是得有 rpm 的管理工具，也就是说得有 rpm 等相关命令;
命令格式： rpm2cpio file.rpm |cpio -div
举例：
[root@localhost RPMS]# rpm2cpio gaim-1.3.0-1.fc4.i386.rpm |cpio -div
抽取出来的文件就在当前操作目录中的 usr,var 和 etc 中;

其实这样抽到文件不如指定安装目录来安装软件来的方便;也一样可以抽出文件;
为软件包指定安装目录：要加 -relocate 参数;下面的举例是把 gaim-1.3.0-1.fc4.i386.rpm 指定安装在 /opt/gaim 目录中;
[root@localhost RPMS]# rpm -ivh --relocate /=/opt/gaim gaim-1.3.0-1.fc4.i386.rpm
Preparing... ### [100%]
1:gaim ### [100%]
[root@localhost RPMS]# ls /opt/
gaim
这样也能一目了然;gaim 的所有文件都是安装在 /opt/gaim 中，我们只是把 gaim 目录备份一下，这样其实也算提取文件的一点用法;

从 file.tgz、file.tar.gz 和 file.tar.bz2 的提取;
[root@localhost ~]# tar zxvf file.tgz
[root@localhost ~]# tar zxvf file.tar.gz
[root@localhost ~]# tar jxvf file.tar.bz2

从 file.deb 的提取;
[root@localhost ~]# ar x file.deb
[root@localhost ~]# tar zxvf data.tar.gz
file.deb 通过 ar x 来解包，然后再把 data.tar.gz 解开就看到相关的目录和文件了;

从 file.iso 文件的提取;
这样的文件在 Linux 主要通过 mount -o loop file.iso 挂载地址;比如 ;
[root@localhost ~]# mount -o loop slackware-10.2-install-d1.iso /mnt/cdrom/

87 ☆硬盘结构

硬盘的物理几何结构是由盘、磁盘表面、柱面、扇区组成，一个张硬盘内部是由几张碟片叠加在一起，这样形成一个柱体面;每个碟片都有上下表面;磁头和磁盘表面接触从而能读取数据;

其中 heads 是磁盘面;sectors 是扇区;cylinders 是柱面;每个扇区大小是 512byte，也就是 0.5K;

所以整个硬盘体积换算公式应该是：
磁面个数 x 扇区个数 x 每个扇区的大小 512 x 柱面个数 = 硬盘体积 （单位 bytes)

首先，硬盘最重要的部分是它的 0 磁面 0 磁道 1 扇区（clindyer 0, side 0, sector 1），在系统 BIOS 自检结束后，如果 BIOS 中定义了首先从硬盘启动，计算机就会把控制权交给了硬盘的 0 磁面 0 磁道 1 扇区，它又叫做“硬盘主引导扇区”。

硬盘主引导扇区由三部分组成，分别是：硬盘主引导记录（MBR）、硬盘分区表（DPT）和结束标志，其总共所占空间为 512 字节（512b），其中硬盘主引导记录（MBR）446 字节（0000--01BD），硬盘分区表（DPT ）64 字节（01BE--01FD），结束标志 2 字节。硬盘主引导记录是用来存放引导程序（Bootloader）的，也就是 Linux 中 Grub 程序放置的地方，负责把操作系统的内核（kernel）读入内存。

硬盘分区表用来存放硬盘分区信息，64 字节被平均分为 4 个部分，也就是每一部分 16 字节，系统用这每 16 个字节记录一个硬盘分区，我们把这样的分区叫做主分区（Primary Partition ）。一个硬盘最多只能有 4 个主分区，这对于计算机的发展是很不利的，随着硬盘容量的扩大这个问题渐渐的明显了，但我们又不能打破以前的分区规则，怎么办呢？我们就把 4 个主分区中的一个拿出来当作特殊的分区处理，在它上面建立新的分区结构。我们把这个特殊的主分区叫做扩展分区（Extended Partition），在扩展分区上划分的新结构分区叫做逻辑分区（Logical Partition）。根据规定 IDE 硬盘可以分 63 个区，SCSI 硬盘可以分 15 个区。

主分区（包括扩展分区）的总个数不能超过四个;也不能把扩展分区包围在主分区之间;
硬盘总容量=主分区（包括扩展分区）总容量
扩展分区容量=逻辑分区总容量

IDE 硬盘能有 4 个主分区,其中一个是扩展分区,总共能有 16 个分区(EL3 中)

88 ☆Linux 存储设备表示方法

IDE 接口硬盘，对于整块硬盘的两种表示方法
一种是 IDE 接口中的整块硬盘在 Linux 系统中表示为/dev/hd[a-z]
另一种表示方法是 hd[0-n] ，其中 n 是一个正整数，比如 hd0,hd1,hd2 ... ... hdn

如果机器中只有一块硬盘，无论我们通过 fdisk -l 列出的是/dev/hda 还是/dev/hdb ，都是 hd0;
如果机器中存在两个或两个以上的硬盘，第一个硬盘/dev/hda 另一种方法表示为 hd0,第二个硬盘/dev/hdb，另一种表法是 hd1

硬盘的分区也有两种表示方法

硬盘分区的第一种表示方法/dev/hd[a-z]X

一种是/dev/hd[a-z]X，这个 a-z 表示 a、b、c.....z ，X 是一个从 1 开始的正整数;比如/dev/hda1，/dev/hda2 .... /dev/hda6，/dev/hda7 ... ... 值得注意的是/dev/hd[a-z]X，如果 X 的值是 1 到 4,表示硬盘的主分区(包含扩展分区);逻辑分区从是从 5 开始的，比如/dev/hda5 肯定是逻辑分区了;

硬盘分区的第二种表示方法(hd[0-n],y)
y 的值是 /dev/hd[a-z]X 中的 X-1 ;
/dev/hda1 等同 (hd0,0)
/dev/hda2 等同 (hd0,1)
/dev/hda5 等同 (hd0,4)
/dev/hda6 等同 (hd0,5)
/dev/hda7 等同 (hd0,6)
/dev/hda8 等同 (hd0,7)
... ...
/dev/hda10 同 (hd0,9)

对于机器中只有一个硬盘来说，无论在 Linux 通过/dev/hda 还是/dev/hdb ，用 hd[0-n]表示方法，都是 hd0;所以如果您如果硬盘中列出来的是;

/dev/hdb1 等同 (hd0,0)
/dev/hdb2 等同 (hd0,1) 注:看好了，这个是扩展分区，在 Linux 还是 Windows 是不能挂载的;
/dev/hdb5 等同 (hd0,4)
/dev/hdb6 等同 (hd0,5)
/dev/hdb7 等同 (hd0,6)
/dev/hdb8 等同 (hd0,7)
... ...
/dev/hdb10 等同 (hd0,9)

注意:如果机器中有两块硬盘，那/dev/hda 另一种表示方法就是 hd0,/dev/hdb 的另一种表示方法是 hd1;这样我们就理解 (hd[0-n],y)的写法了吧;这样机器只有单个硬盘或者多个硬盘，我们都知道怎么写了;

关于 SATA 和 SCSI 接口的硬盘的两种表示方法

理解方法和 IDE 接口的硬盘相同，只是把 hd 换成 sd;

如果您的机器中比如有一个硬盘是/dev/hda ，也有一个硬盘是/dev/sda ，那/dev/sda 的硬盘应该是 sd0; 具体每个分区用(sd[0-n],y)的表示方法和 IDE 接口中的算法相同，比如/dev/sda1 就是(sd0,0);

usb 及 1394 接口的存储设备和软驱设备;

usb 存储设备也目前在内核中在两种驱动方法，一种是模拟 SCSI 硬盘，通过 fdisk -l 出现的是 /dev/sd[0-n] ;如果是模拟 SCSI 设备的方法来驱动。那 usb 存储设备在 Linux 的另一种表示方法和前面所说的 SCSI 和 SATA 的相同;

但目前新版本的内核中，想抛弃模拟 SCSI，我们通过 fdisk 列系统存在的存储设置时会出现 /dev/uba 类似的;但目前这个驱动并不成熟,比如大数据量表现不稳定;其实 USB 接口的存储设备，在 Linux 表现还是比较差;

1394 接口存储调备，在 Linux 中也是模拟 SCSI，我们通过 fdisk -l 后，出现的也是/dev/sd[0-n]，另一种表示方法(sd[a-z],y)的理解请参照前面所说的; 1394 接口的存储设备在 Linux 表现极好，USB 存储如果相对 1394 接口的存储表现来说，应该不值不提，建议大家购买 1394 接口的存储设备;

软驱在 Linux 中，是/dev/fd0 设备这是一般情况，另一种表示为 fd0;

CDROM 或 DVDROM ，以及 COMBO ，一般的情况下是/dev/hdc ;看下面的例子，无论是 /dev/cdrom 还是/dev/dvd ，最后都指向了/dev/hdc;

```
[root@localhost ~]# ls -la /dev/cdrom
lrwxrwxrwx 1 root root 3 2005-12-14 /dev/cdrom -> hdc
[root@localhost ~]# ls -la /dev/dvd
lrwxrwxrwx 1 root root 3 2005-12-14 /dev/dvd -> hdc
```

FreeBSD 中

(hd0,2,a): 专用于 FreeBSD,FreeBSD 有一个 slice 概念，把一个分区进一步分为几个 slice,此处指明是第一块硬盘的第三个分区中的 slice a。你也可以用(hd0,a)，这样 GRUB 就会在第一块硬盘上找到第一个 FreeBSD 分区的 slice a。

89 ☆fdsik

fdisk 是一款强大的磁盘操作工具，来自 util-linux 软件包

fdisk 操作硬盘的命令格式如下：
[root@panda ~]# fdisk 设备

[root@panda ~]# fdisk /dev/hda
或
[root@panda ~]# fdisk /dev/sda

fdisk /dev/sdb
进入 fdisk
m 帮助
d 删除分区
n 添加分区
t 转换分区类型
l 列出支持的分区类型
p 显示分区信息
w 写入分区表
q 退出不保存
v 校验分区表,查看有没有没有分配的空间

[root@panda ~]# fdisk /dev/sda
Command (m for help): 在这里按 m ，就会输出帮助;
Command action
a toggle a bootable flag
b edit bsd disklabel
c toggle the dos compatibility flag
d delete a partition 注：这是删除一个分区的动作;
l list known partition types 注：l 是列出分区类型，以供我们设置相应分区的类型;
m print this menu 注：m 是列出帮助信息;
n add a new partition 注：添加一个分区;
o create a new empty DOS partition table
p print the partition table 注：p 列出分区表;列出当前操作硬盘的分区情况
q quit without saving changes 注：不保存退出;
s create a new empty Sun disklabel
t change a partition's system id 注：t 改变分区类型;
u change display/entry units
v verify the partition table
w write table to disk and exit 注：把分区表写入硬盘并退出;
x extra functionality (experts only) 注：扩展应用，专家功能;

通过-l 参数，能获得机器中所有的硬盘的分区情况,也能列出机器中所有磁盘的个数
[root@panda ~]# fdisk -l
Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	522	4088542+	8e	Linux LVM

Disk /dev/sdb: 107 MB, 107374080 bytes
64 heads, 32 sectors/track, 102 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	1	102	104432	8e	Linux LVM

请注意第一行， Disk /dev/sdf: 107 MB, 107374080 bytes ，这个就是表示机器中有一个硬盘设备 /dev/hda ，体积大小为 107 MB;下面的就是硬盘的分区，每个分区都有详细的信息，
在上面 Blocks 中，表示的是分区的大小，Blocks 的单位是 byte
System 表示的文件系统类型

也可以指定 fdisk -l 来查看其中一个硬盘的分区情况;只能为磁盘号,不能为分区号
[root@localhost ~]# fdisk -l /dev/sda

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	522	4088542+	8e	Linux LVM

如:
Disk /dev/hda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

这个硬盘是 80G 的，有 255 个磁面;63 个扇区;9729 个磁柱;每个 cylinder（磁柱）的容量是 8225280 bytes=8225.280 K（约为）=8.225280M（约为）;

分区序列 引导 开始 终止 容量 分区类型 ID 分区类型
Device Boot Start End Blocks Id System
/dev/hda1 * 1 765 6144831 7 HPFS/NTFS
/dev/hda2 766 2805 16386300 c W95 FAT32 (LBA)
/dev/hda3 2806 9729 55617030 5 Extended
/dev/hda5 2806 3825 8193118+ 83 Linux
/dev/hda6 3826 5100 10241406 83 Linux
/dev/hda7 5101 5198 787153+ 82 Linux swap / Solaris
/dev/hda8 5199 6657 11719386 83 Linux
/dev/hda9 6658 7751 8787523+ 83 Linux
/dev/hda10 7752 9729 15888253+ 83 Linux

说明：
硬盘分区的表示：在 Linux 是通过 hd*x 或 sd*x 表示的，其中 * 表示的是 a、b、c ... ... x 表示的数字 1、2、3 ... ... hd 大多是 IDE 硬盘;sd 大多是 SCSI 或移动存储;
引导（Boot）：表示引导分区，在上面的例子中 hda1 是引导分区;
Start （开始）：表示的一个分区从 X cylinder（磁柱）开始;
End （结束）：表示一个分区到 Y cylinder（磁柱）结束;
id 和 System 表示的是一个意思，id 看起来不太直观，我们要在 fdisk 一个分区时，通过指定 id 来确认分区类型;比如 7 表示的就 NTFS 分区;这个在 fdisk 中要通过 t 功能来指定。下面的部份会提到;
Blocks（容量）：这是我翻译的，其实不准确，表示的意思的确是容量的意思，其单位是 K;一个分区容量的值是由下面的公式而来的;

Blocks = （相应分区 End 数值 - 相应分区 Start 数值）x 单位 cylinder（磁柱）的容量

所以我们算一下 hda1 的 Blocks 的大小 ：
hda1 Blocks=(765-1)x8225.280=6284113.92 K = 6284.113.92M

注：换算单位以硬盘厂家提供的 10 进位算起，如果以操作系统二进制来算，这个分区容量应该更少一些，得出的这个值和我们通过 fdisk -l 看到的 /dev/hda1 的值是大体相当的，因为换算方法不一样，所以也不可能尽可能的精确;再加上分区时的一点损失之类，有时或大或小是存在的;

我们查看分区大小或者文件的时候，还是用十进制来计算比较直观;推算办法是 byte 向前推小数点三位就是 K ， K 单位的值向前推小数点三位就是 M，M 向前推小数点三位就是 G... ... 一般也差不了多少;这么算就行;

90 ☆parted

parted 默认是打开的设备是/dev/hda ，也可以自己指定;比如 parted /dev/hda 或/dev/sda 等;退出的方法是 quit

[root@panda ~]# parted
使用 /dev/hda
(parted) p
/dev/hda 的磁盘几何结构:0.000-76319.085 兆字节
磁盘标签类型:msdos
Minor 起始点 终止点 类型 文件系统 标志
1 0.031 6000.842 主分区 ntfs 启动
2 6000.842 22003.088 主分区 fat32 lba
3 22003.088 60800.690 扩展分区
5 22003.119 30004.211 逻辑分区 reiserfs
6 30004.242 40005.615 逻辑分区 reiserfs
7 40005.646 40774.350 逻辑分区 linux-swap
8 40774.381 52219.094 逻辑分区 ext3
9 52219.125 60800.690 逻辑分区 reiserfs

我们在 partd 的操作面上，用 p 就可以列出当前磁盘的分区情况，如果想要查看其它磁盘，可以用 select 功能，比如 select /dev/sda ;

实际操作如下:
(parted) print
/dev/hdb 的磁盘几何结构：0.000-38172.750 兆字节
磁盘标签类型：msdos
Minor 起始点 终止点 类型 文件系统 标志
1 0.031 3498.530 主分区 ntfs 启动
2 3498.530 32788.916 扩展分区 lba
5 3498.561 9499.372 逻辑分区 fat32
6 9499.403 19500.776 逻辑分区 ext3
7 19500.807 25399.643 逻辑分区 ext3
8 25399.674 25784.011 逻辑分区 linux-swap
9 25784.042 32788.916 逻辑分区 ext3
(parted)
这样就可以看到分区情况了，起始点和中止点的单位是 MB,创建分区的时候会用到
(parted) mkpart
分区类型？ primary/主分区/logical/逻辑分区? p
文件系统类型？ [ext2]? ext3
起始点？ 32789
结束点？ 33000
(parted)
创建了一个主分区，大小是 33000-32789=211。为主分区。如果你的硬盘里面没有扩展分区，那么它会提示主分区和扩展分区，你可以从中选择

还有一个参数 mkpartfs,我没发现他们有什么不同。不同好象就是在输入 fs 的时候如果我输入 ext3,那么 mkpartfs 就不通过，说不支持。可是实际上 mkpart 即使输入了 ext3,实际创建的还是 ext2。
还有一个参数 set
(parted) set 3
要修改的标志？ lvm
新状态？ 开/on/[关]/off? on

91 ☆sfdisk

列磁盘分区情况的功能
[root@panda ~]# sfdisk -l

92 ☆partx

用法: partx 设备名

[root@localhost ~]# partx /dev/sda
1: 63- 208844 (208782 sectors, 106 MB)
2: 208845- 8385929 (8177085 sectors, 4186 MB)
3: 0- -1 (0 sectors, 0 MB)
4: 0- -1 (0 sectors, 0 MB)

93 ☆mkfs

使用方法:
[root@panda ~]# mkfs -t 文件系统 存储设备
注：
这里的文件系统是要指定的，比如 ext3 ;reiserfs ;ext2 ;fat32 ;msdos 等... ...
设备比如是一个硬盘的分区，软盘，光驱等.. ...

在格式化分区之前，您得懂得如何查看硬盘分区情况，并有针对性的格式化;比如用 fdisk -l 来查看;

[root@panda ~]# mkfs -t ext3 /dev/sdb1
[root@panda ~]# mkfs -t ext2 /dev/sda1
[root@panda ~]# mkfs -t reiserfs /dev/sda1
[root@panda ~]# mkfs -t fat32 /dev/sda1
[root@panda ~]# mkfs -t msdos /dev/sda1

也可以为:
mkfs.类型 分区

用 mkfs.bfs mkfs.ext2 mkfs.jfs mkfs.msdos mkfs.vfatmkfs.cramfs mkfs.ext3 mkfs.minix mkfs.reiserfs mkfs.xfs 等命令来格式化分区
[root@panda ~]# mkfs.ext3 /dev/sda1 注：把该设备格式化成 ext3 文件系统
[root@panda ~]# mkfs.ext2 /dev/sda1 注：把该设备格式化成 ext2 文件系统
[root@panda ~]# mkfs.reiserfs /dev/sda1 注：把该设备格式化成 reiserfs 文件系统
[root@panda ~]# mkfs.vfat /dev/sda1 注：把该设备格式化成 fat32 文件系统
[root@panda ~]# mkfs.msdos /dev/sda1 注：把该设备格式化成 fat16 文件系统,msdos 文件系统就是 fat16;
[root@panda ~]# mkdosfs /dev/sda1 注：把该设备格式化成 fat16 文件系统，同 mkfs.msdos

mke2fs
[root@panda ~]# mke2fs /dev/sda1 注：把该设备格式化成 ext2 文件系统
[root@panda ~]# mke2fs -j /dev/sda1 注：把该设备格式化成 ext3 文件系统

创建时使用 2k 的块，每 4k 一个 inode 的设置
mke2fs -b 2048 -i 8096 /dev/sda1
缺省为 ext2,除非加上-j

94 ☆mkswap

用磁盘分区作为交换分区:
[root@panda ~]# mkswap /dev/sdb1 注：创建此分区为 swap 交换分区
[root@panda ~]# swapon /dev/sdb1 注：加载交换分区;
[root@panda ~]# swapoff /dev/sdb1 注：关闭交换分区;

[root@panda ~]# swapon -s 注:查看系统已经加载的 swap 交换分区;

Filename	Type	Size	Used	Priority
/dev/mapper/VolGroup00-LogVol01	partition	524280	0	-1
/dev/sdb1	partition	51176	0	-3

如果让 swap 开机就加载，应该改 /etc/fstab 文件，加类似如下一行;
/dev/sdb1 swap swap defaults 0 0 注：把此行中的/dev/hda7 改为您的交换分区就行;
或者把命令行直接写入 /etc/rc.d/rc.local 中也行;
swapon /dev/sdb1

用文件做交换分区:
[root@panda ~]# dd if=/dev/zero of=/tmp/swap bs=1024 count=262144
注：创建一个大小为 256M 的 swap 文件，在/tmp 目录中;您可以根据自己的需要的大小来创建 swap 文件;

[root@panda ~]# mkswap /tmp/swap
注：把/tmp/swap 文件，创建成 swap 交换区

Setting up swapspace version 1, size = 536866 kB
no label, UUID=d9d8645d-92cb-4d33-b36e-075bb0a2e278
[root@panda ~]# swapon /tmp/swap
注：挂载 swap
[root@panda ~]# swapon -s
Filename Type Size Used Priority
/dev/hda7 partition 787144 888 -1
/tmp/swap file 524280 0 -2

注意：其实我们在安装系统的时候，就已经划分了交换分区;查看/etc/fstab，应该 swap 的行;如果您在安装系统时没有添加 swap，可以通过这种办法来添加;

95 ☆e2label

e2label device [newlabel]
设置设备标签
e2label /dev/sdb1 /data

96 ☆tune2fs

tune2fs 设备名(只能是设备分区名,不能为挂载点)
-l [设备] :看文件系统信息
-c [次数] :强制挂载次数自检
-i [天数] :强制自检的间隔时间
-m [百分比] :保留块百分比,可修改,不一定格式化才改
-j :转换 ext2 到 ext3

显示文件系统的 characteristics
tune2fs -l /dev/<partition>
分区属性:
inode count
block count
来看分区的节点总数和块总数
节点和块数接近比例由-i : -b 来决定
保留块为管理员保留的磁盘管理空间默认保留块为总数据块的 5%,硬盘越大就越浪费
mkfs.ext3 -b 4096 -i 8192 -m 2 /dev/sda2
可设置为 2%

转换前用 sync
tune2fs -j -c 0 -i 0 /dev/<partition>

tune2fs -c -1 /dev/sdb1

永远不会自检,mount count 永远也到达不了 maximum mount count

tune2fs -i 0 /dev/sdb1
0 永远不会自检

tune2fs -m 10 /dev/sdb1
改保留块百分比

97 ☆reiserfstune

[root@panda ~]# reiserfstune -l 标签 设备

举例：比如我为 reiserfs 文件系统 /dev/hda10 设置标签为 /10 ;
[root@panda ~]# reiserfstune -l /10 /dev/hda10

我们在/etc/fstab 中加入一行;
/10 /mnt/hda10 reiserfs defaults 0 0

98 ☆fsck

fsck 设备名(只能是设备分区名,不能为目录)
文件系统的检查工具
注意类型,不要混用

在 init 1 下
umount /mnt/new
fsck -t ext3 /dev/sdb1
mount /dev/sdb1 /mnt/new

值得注意的是 fsck 扫描文件系统时一定要在单用户模式、修复模式或把设备 umount 后进行。
警告：如果扫描正在运行中的系统，会造成系统文件损坏;如果您的系统是正常，请不要用扫描工具，她可能会把您的系统搞坏掉，fsck 运行是有危险的;

对于 fsck.ext2 和 fsck.ext3 常用的几个选项：
-p Automatic repair (no questions) 注：自动修复文件系统存在的问题;
-y Assume "yes" to all questions 注：如果文件系统有问题，会跳出提示是否修复，如果修复请按 y;
-c Check for bad blocks and add them to the badblock list 注：对文件系统进行坏块检查;这是一个极为漫长的过程;
-n Make no changes to the filesystem 注：不对文件系统做任何改变，只要扫描，以检测是否有问题;

[root@panda ~]# fsck.ext3 -p /dev/sdb1

注意： 针对不同文件系统，最好用相应的工具;虽然有时 fsck 在不加参数的情况下能识别不同的文件系统;

以 Fedora 为例，文件系统扫描工具有 fsck fsck.ext2 fsck.jfs fsck.msdos fsck.vfat fsck.ext3 fsck.reiserfs（reiserfsck）

其中 fsck 默认支持文件系统 ext2，如果想支持 ext3 文件系统的扫描，应该加-j 参数，最好是我们应该根据不同的文件系统来调用不同的扫描工具，比如 fsck.ext2，fsck.jfs，fsck.msdos，fsck.ext3，fsck.reiserfs（reiserfsck）等。我们也可以根据自己的文件系统选择不同的扫描工具;

e2fsck 专门检查 ext2/ext3
很危险的!!!

e2fsck /dev/hdx 修复文件系统

99 ☆mount

功能：加载指定的文件系统。
语法：mount [-afFhnrvVw] [-L<标签>] [-o<选项>] [-t<文件系统类型>] [设备名] [加载点]
用法说明：mount 可将指定设备中指定的文件系统加载到 Linux 目录下（也就是装载点）。可将经常使用的设备写入文件/etc/fastab,以使系统在每次启动时自动加载。mount 加载设备的信息记录在/etc/mtab 文件中。使用 umount 命令卸载设备时，记录将被清除。
常用参数和选项：
-a 加载文件/etc/fstab 中设置的所有设备。
-f 不实际加载设备。可与-v 等参数同时使用以查看 mount 的执行过程。
-F 需与-a 参数同时使用。所有在/etc/fstab 中设置的设备会被同时加载，可加快执行速度。
-h 显示在线帮助信息。
-L<标签> 加载文件系统标签为<标签>的设备。
-n 不将加载信息记录在/etc/mtab 文件中。
-o<选项> 指定加载文件系统时的选项。有些选项也可在/etc/fstab 中使用。这些选项包括：
async 以非同步的方式执行文件系统的输入输出动作。
atime 每次存取都更新 inode 的存取时间，默认设置，取消选项为 noatime。
auto 必须在/etc/fstab 文件中指定此选项。执行-a 参数时，会加载设置为 auto 的设备，取消选取为 noauto。
defaults 使用默认的选项。默认选项为 rw、suid、dev、exec、auto nouser 与 async。
dev 可读文件系统上的字符或块设备，取消选项为 nodev。
exec 可执行二进制文件，取消选项为 noexec。
noatime 每次存取时不更新 inode 的存取时间。
noauto 无法使用-a 参数来加载。
nodev 不读文件系统上的字符或块设备。
noexec 无法执行二进制文件。
nosuid 关闭 set-user-identifier(设置用户 ID)与 set-group-identifer(设置组 ID)设置位。

nouser 使一位用户无法执行加载操作，默认设置。
remount 重新加载设备。通常用于改变设备的设置状态。
ro 以只读模式加载。
rw 以可读写模式加载。
suid 启动 set-user-identifier(设置用户 ID)与 set-group-identifer(设置组 ID)设置位，取消选项为 nosuid。
sync 以同步方式执行文件系统的输入输出动作。
user 可以让一般用户加载设备。

-r 以只读方式加载设备。
-t<文件系统类型> 指定设备的文件系统类型。常用的选项说明有：
minix Linux 最早使用的文件系统。
ext2 Linux 目前的常用文件系统。
msdos MS-DOS 的 FAT。
vfat Win85/98 的 VFAT。
nfs 网络文件系统。
iso9660 CD-ROM 光盘的标准文件系统。
ntfs Windows NT 的文件系统。
hpfs OS/2 文件系统。Windows NT 3.51 之前版本的文件系统。
auto 自动检测文件系统。

-v 执行时显示详细的信息。
-V 显示版本信息。
-w 以可读写模式加载设备，默认设置。

查看分区是否被挂载了的命令，直接用 mount -s
[root@panda ~]# mount -s

观察挂载选项
ro/rw
exec/noexec 可执行文件被执行
dev/nodev 辨认设备文件
suid,sgid/nosuid,nosgid 设置 suid,sgid 文件是否生效
atime/noatime 更新节点的访问时间
async/sync 异步或同步
user/nouser 普通用户挂载磁盘 ,只能在 fstab 文件内使用
默认都是可以,逗号隔开
在 fstab 中的 default 那一列使用
default 表示 rw,suid,dev,exec,auto,nouser,async

设备

指存储设备，至于您的系统中有哪些存储设备，主要通过 fdisk -l 或者查看 /etc/fstab 或 dmesg ; 一般的情况下光驱设备是/dev/cdrom ;软驱设备是/dev/fd0 ;硬盘及移动硬盘以 fdisk -l 的输出为准;

挂载设备过程:
[root@panda ~]# mkdir /mnt/sda6
[root@panda ~]# chmod 777 /mnt/sda6 #打开权限，所有用户可读可写可执行;
[root@panda ~]# mount /dev/sda1 /mnt/sda6

mount /dev/sdb1 /mnt/new
路径最后不要加/

mount /mnt/new -o remount,ro
现在是只读的了

mount /dev/sdb1 /mnt/d1 -o noexec
不要把根分区挂载成 noexec,所有程序都无法启动,在防止移动介质上的有害程序,才使用

在移动介质上可能有设备文件,可以通过设备文件访问你实际的硬盘,很危险的,在共享目录中也要注意
suid,sgid 执行者可以获得程序运用者权限,危险,共享目录中注意
atime 频繁的访问文件,更新的开销会很大
sync 在实时中用,async 可以减少 cpu 开销

100 ☆umount

umount 设备或挂载目录

101 ☆fstab

/etc/fstab
第一字段：设备名，在这里表示是文件系统; 有时我们把挂载文件系统也说成挂载分区; 在这个字段中也可以用分区标签;在例子中/LABEL=/1 就是 Fedora 系统安装分区的标签，至于是在哪个分区，可以用 df -lh 来查看;我们可以知道 LABEL=/1 是/dev/hda8 的标签;
第二字段：文件系统的挂载点;
第三字段：文件系统类型;
第四字段：mount 命令的选项，和 mount 中的-o 同理;defaults 包括这些选项 rw, suid, dev, exec, auto, nouser, async;通过实践，这个默认的还能满足我们的需要;至于这些选项的意思，请参看 man mount ;
第五字段：表示文件系统是否需要 dump 备份，是真假关系;1 是需要，0 是不需要;
第六字段：是否在系统启动时，通过 fsck 磁盘检测工具来检查文件系统，1 是需要，0 是不需要，2 是跳过;
只有根分区的自检顺序是 1

其他可有多个同顺序号的

LABEL=/data /data ext2 defaults 1 2
或者
/dev/hdax /data ext2 defaults 1 2
以上两行的实际效果相同。但是，如果更换了硬盘的总线或者是更改了主从的顺序，在 fstab 中使用卷标仍然可以定位这个设备
LABEL=/data　　/data(注意是目录)　　ext2 defaults　1 2

mount /data
就是直接用 fstab 文件,至于 mount 到哪了，我们可以通过查看/etc/fstab 来查看

dvd 、cdrom 、cdwriter 的文件名都链接到了 hdc 这个设备，所以光驱设备根源就是/dev/hdc

102☆df

df 是来自于 coreutils 软件包，系统安装时，就自带的;我们通过这个命令可以查看磁盘的使用情况以及文件系统被挂载的位置

功能:检查文件系统的磁盘空间占用情况。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。

语法:
df [选项]

该命令各个选项的含义如下:
-a 显示所有文件系统的磁盘使用情况，包括 0 块(block)的文件系统，如/proc 文件系统。
-k 以 k 字节为单位显示。
-i 显示 i 节点信息，而不是磁盘块。
-t 显示各指定类型的文件系统的磁盘空间使用情况。
-x 列出不是某一指定类型文件系统的磁盘空间使用情况(与 t 选项相反)。
-T 显示文件系统类型。
-h 除 1024
-H 除 1000

[root@panda ~]# df -lh (常用)
Filesystem　　Size　Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
　　3.6G　2.2G　1.3G　63% /
/dev/sda1　　99M　 10M　 84M　11% /boot
none　　126M　　0　126M　 0% /dev/shm
/dev/sdd1　　101M　8.4M　 87M　 9% /data

列出各文件系统的磁盘空间使用情况。

```
$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                       3773552   2254752   1327108  63% /
/dev/sda1              101086     10204     85663  11% /boot
none                    128020         0    128020   0% /dev/shm
/dev/sdd1              102704      8538     88948   9% /data
```

df 命令的输出清单的第 1 列是代表文件系统对应的设备文件的路径名(一般是硬盘上的分区);第 2 列给出分区包含的数据块(1024 字节)的数目;第 3，4 列分别表示已用的和可用的数据块数目。用户也许会感到奇怪的是，第 3，4 列块数之和不等于第 2 列中的块数。这是因为缺省的每个分区都留了少量空间供系统管理员使用。即使遇到普通用户空间已满的情况，管理员仍能登录和留有解决问题所需的工作空间。清单中 Use% 列表示普通用户空间使用的百分比，即使这一数字达到 100%，分区仍然留有系统管理员使用的空间。最后，Mounted on 列表示文件系统的安装点。

列出各文件系统的 i 节点使用情况。

```
$ df -ia
Filesystem           Inodes   IUsed   IFree IUse% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      480000  113526  366474   24% /
none                       0       0       0    -  /proc
none                       0       0       0    -  /sys
none                       0       0       0    -  /dev/pts
usbfs                      0       0       0    -  /proc/bus/usb
/dev/sda1                  26104   35      26069    1% /boot
none                       32005   1       32004    1% /dev/shm
/dev/sdd1                  13248   12      13236    1% /data
none                       0       0       0    -  /proc/sys/fs/binfmt_misc
sunrpc                     0       0       0    -  /var/lib/nfs/rpc_pipefs
automount(pid2513)         0       0       0    -  /misc
```

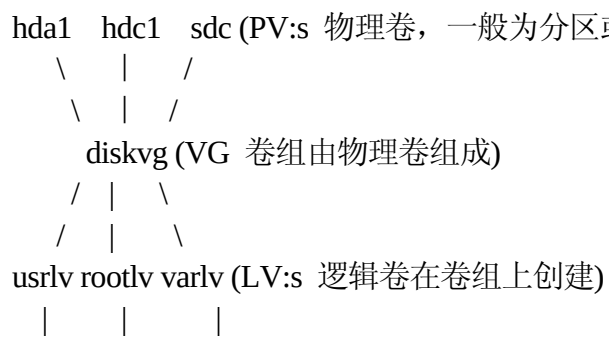
列出文件系统的类型。

```
$ df -T
/dev/mapper/VolGroup00-LogVol00
                     ext3      3773552   2254752   1327108  63% /
/dev/sda1            ext3       101086     10204     85663  11% /boot
none                 tmpfs       128020         0    128020   0% /dev/shm
/dev/sdd1            ext3       102704      8538     88948   9% /data
```

103☆LVM

LVM 是 Logical Volume Manager(逻辑卷管理)的简写，它由 Heinz Mauelshagen 在 Linux 2.4 内核上实现

LVM 的结构



物理块 physical extent (PE)
物理卷按大小相等的"块"为单位存储，块的大小与卷组中逻辑卷块的大小相同。

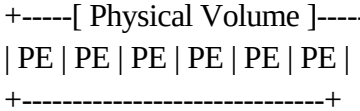
逻辑块 logical extent (LE)
逻辑卷按"块"为单位存储，在一卷组中的所有逻辑卷的块大小是相同的。

物理卷 physical volume (PV)
典型的物理卷是硬盘分区，但也可以是整个硬盘或已创建的 Software RAID 卷。

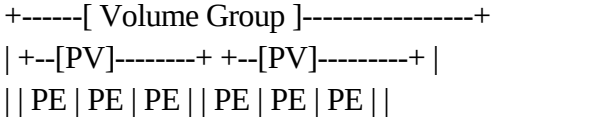
卷组 volume group (VG)
卷组是 LVM 中最高抽象层，是由一个或多个物理卷所组成的存储器池。

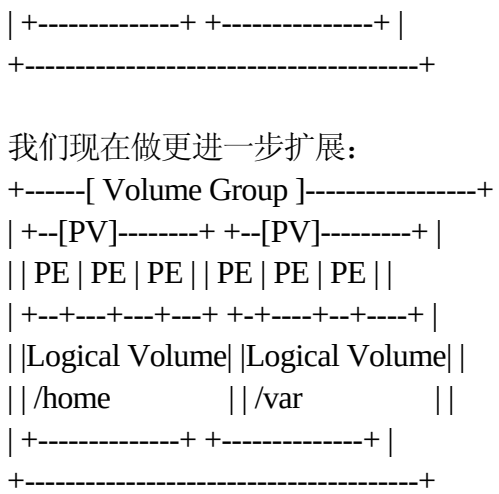
逻辑卷 logical volume (LV)
逻辑卷相当于非 LVM 系统中的分区，它在卷组上建立，是一个标准的块设备，可以在其上建立文件系统。

一个物理卷，包含了许多物理分区：



一个卷组，包含了 2 个物理卷（PV）有 6 个物理分区：





我们现在做更进一步扩展：

映射模式 (linear/striped)
在建立逻辑卷时，可以选择逻辑块与物理块映射的策略:
1.线性映射-将把一定范围的物理块按顺序分配给逻辑卷，如 LV 的 LE 1 - 99 映射到 PV1，LE 100 - 347 映射到 PV2。
2.交错模式-将把逻辑块交错映射到不同的物理卷中，如 LV 的 LE 1 映射为 PV1 的 PE1，LE 2 映射为 PV2 的 PE1，LE 3 映射为 PV1 的 PE2...。这种方式可以提高逻辑卷的性能，但是采用这种方式建立的逻辑卷将不能在它们所在的物理卷中扩展。

Snapshots (快照)
LVM 提供了一个非常好的特性:snapshots。它允许管理员建立一个块设备:该设备是一逻辑卷在某一时刻冻结的精确拷贝。这个特性通常用于批处理过程(如备份)需要处理逻辑卷，但又不能停止系统。当操作完成时，snapshot 设备可以被移除。这个特性要求在建立 snapshot 设备时逻辑卷处于相容状态。

104 ☆LVM 建立 PV

PV 初始化命令 pvcreate 的一般用法为:
pvcreate PV1 [PV2 ...]
它的参数可以是整个磁盘、分区，也可以是一 loop 设备。

为把一个磁盘或分区作为 PV，首先应使用 pvcreate 对其初始化，如对 IDE 硬盘/dev/hdb，

使用整个磁盘，
pvcreate /dev/hdb
这将在磁盘上建立 VG 的描述符。先用 fdisk 分区

使用磁盘分区，如/dev/hdb1。
使用 fdisk 的 t 命令把/dev/hda1 的分区类型设为 0x8e，然后运行:
pvcreate /dev/hdb1

这将在分区/dev/hda1 上建立 VG 的描述符。
pvcreate 在每个硬盘的起始端建立卷组描述区(volume group descriptor area, VGDA)。

105 ☆LVM 建立 VG

VG 初始化命令 vgcreate 的一般用法为:
vgcreate [options] VG_NAME PV1 [PV2 ...]
其中的可选项包括设置 VG 最大支持的 LV 数、PE 大小(缺省为 4MB)等。

在使用 pvcreate 建立了 PV 后，可以用 vgcreate 建立卷组，如有 PV1、PV2 分别是/dev/hda1 与/dev/hdb1，使用
vgcreate testvg /dev/sdb1 /dev/sdc
将建立一个名为 testvg 的卷组，它由两个 PV:/dev/hda1 与/dev/hdb1 组成。

注意:当使用 devfs 系统时，应使用设备的全名而不能是 Symbol Link，如对上例应为:
vgcreate testvg /dev/ide/host0/bus0/target0/lun0/part1 /dev/ide/host0/bus0/target1/lun0/part1

106 ☆LVM 激活 VG

激活卷组命令 vgchange 可用来设置 VG 的一些参数
vgchange -a [y|n] test_vg
如是否可用(-a [y|n]选项)、支持最大逻辑卷数等。

在被激活之前，VG 与 LV 是无法访问的，这时可用命令:
vgchange -a y testvg
激活所要使用的卷组。当不再使用 VG 时，可用
vgchange -a n testvg
使之不再可用。
对卷组的操作都应该先休眠卷组

107 ☆LVM 移除 VG

在移除一卷组前应确认卷组中不再有逻辑卷 LV，首先休眠卷组:
vgchange -a n testvg
然后可用 vgremove 移除该卷组:
vgremove testvg

108 ☆LVM 为 VG 增加 PV

当卷组空间不足时，可以加入新的物理卷来扩大容量，这时可用命令 vgextend，如
vgextend testvg /dev/sdb5
其中/dev/hdc1 是新的 PV，当然在这之前，它应使用 pvcreate 初始化。

109 ☆LVM 从 VG 移除 PV

在移除 PV 之前，应确认该 PV 没用被 LV 使用，这可用命令 pvdisplay 查看，如:
pvdisplay /dev/hda1
--- Physical volume ---
PV Name /dev/hda1
VG Name testvg
PV Size 1.95 GB / NOT usable 4 MB [LVM: 122 KB]
PV# 1
PV Status available
Allocatable yes (but full)
Cur LV 1
PE Size (KByte) 4096
Total PE 499
Free PE 0
Allocated PE 499
PV UUID Sd44tK-9IRw-SrMC-MOkn-76iP-iftz-OVSen7

注意观察"Allocated PE"这一项,为 0 代表没有被 lv 使用
如这个 PV 仍在被使用，则应把数据传移到其它 PV 上。在确认它未被使用后，可用命令 vgreduce 把它从 VG 中删除，如:
vgreduce testvg /dev/sdb5

110 ☆LVM 在 PV 间转移数据

若要把一个 PV 从 VG 中移除，应首先把其上所有活动 PE 中的数据转移到其它 PV 上，而新的 PV 必须是本 VG 的一部分，有足够的空间。如要把 PV1:/dev/hda1 上的数据移到 PV2:/dev/sda1 上可用命令:
pvmove /dev/sdb1 /dev/sdc
如果在该 PV 之上的 LV 采用交错方式存放，则这个转移过程不能被打断。
建议在转移数据之前备份 LV 中的数据。

111 ☆LVM 创建 LV

命令 lvcreate 的常用方法:
lvcreate [options] -n 逻辑卷名 卷组名 [PV1 ...]
其中的常用可选项有:
"-i Stripes :采用交错(striped)方式创建 LV，其中 Stripes 指卷组中 PV 的数量。注意:如果使用 -i2 参数，则 LV 将仅使用 test_vg 中的两块硬盘。
"-I Stripe_size :采用交错方式时采用的块大小(单位为 KB)，Stripe_size 必须为 2 的指数:2N ，N=2,3...9。

"-l LEs :指定 LV 的逻辑块数。
"-L size :指定 LV 的大小，其后可以用 K、M、G 表示 KB、MB、GB。
"-s :创建一已存在 LV 的 snapshot 卷。
"-n name :为 LV 指定名称。
在刚创建 VG 时 dev 中不会出现以 VG 名为名的文件夹，但是当创建了 LV 以后，则会出现，并且在文件夹中有一个以 LV 名命名的文件

在创建逻辑卷前，应决定 LV 使用哪些 PV，这可用命令 vgdisplay 与 pvdisplay 查看当前卷组与 PV 的使用情况。在已有的卷组上创建逻辑卷使用命令 lvcreate，如:
lvcreate -L1500 -n testlv testvg
将在卷组 testvg 上建立一个 1500MB 的线性 LV，其命名为 testlv，对应的块设备为/dev/testvg/testlv。
lvcreate -i2 -I4 -l100 -n anothertestlv testvg
将在卷组 testvg 上建立名为 anothertestlv 的 LV，其大小为 100LE，采用交错方式存放，交错值为 2(2 个 pv 上)，块大小为 4KB。

如果需要 LV 使用整个 VG，可首先用 vgdisplay 查找 Total PE 值，然后在运行 lvcreate 时指定，如:
vgdisplay testvg | grep "Total PE"
Total PE 10230
lvcreate -l 10230 testvg -n testlv
将使用卷组 testvg 的全部空间创建逻辑卷 testvg。

在创建逻辑卷后，就可在其上创建文件系统并使用它。
逻辑卷上创建文件系统:mkreiserfs /dev/test_vg/testlv

这个设备是个连接
/dev/vg01/lv01 -> /dev/mapper/vg01-lv01

112 ☆LVM 删除 LV

为删除一个逻辑卷，必须首先从系统卸载其上的文件系统，然后可用 lvremove 删除，如:
umount /dev/testvg/testlv
lvremove /dev/testvg/testlv #必须指定全路径
lvremove -- do you really want to remove "/dev/testvg/testlv"? [y/n]: y
lvremove -- doing automatic backup of volume group "testvg"
lvremove -- logical volume "/dev/testvg/testlv" successfully removed

113 ☆LVM 扩展 LV

为逻辑卷增加容量可用使用 lvextend，即可以指定要增加的尺寸也可以指定扩容后的尺寸，如
lvextend -L12G /dev/testvg/testlv #必须全路径
lvextend -- extending logical volume "/dev/testvg/testlv" to 12 GB

lvextend -- doing automatic backup of volume group "testvg"
lvextend -- logical volume "/dev/testvg/testlv" successfully extended

将扩大逻辑卷 testlv 的容量为 12GB。
lvextend -L+1G /dev/testvg/testlv
lvextend -- extending logical volume "/dev/testvg/testlv" to 13 GB
lvextend -- doing automatic backup of volume group "testvg"
lvextend -- logical volume "/dev/testvg/testlv" successfully extended
将为 LV testlv 再增大容量 1GB 至 13GB。

为 LV 扩容的一个前提是:LV 所在的 VG 有足够的空闲存储空间可用。
在为 LV 扩容之后，应同时为 LV 之上的文件系统扩容，使二者相匹配。对不同的文件系统有相对应的扩容方法。

(推荐使用工具)
ext2online device size (MmKG)
ext2online 工具被添加用来在线地扩大已存在的 ext3 文件系统。
备注
需要注意的是，ext2online 并不能扩大它所在的块设备本身．一定要有足够的未被使用的空间在这个设备上。最简单的方法是使用 LVM 卷并运行 lvresize 或 lvextend 来扩展这个设备。
[root@client ~]# lvextend -L +5M /dev/testvg/testlv
先扩大 LV
[root@client ~]# df -h /data
大小没有变化
[root@client ~]# ext2online -d -v /dev/testvg/testlv　　#不用 umount
再扩大文件系统,不用 umount
[root@client ~]# df -h /data
在你 lvextend -L +xxxM 之后
df -Th /mountpoint 看一下 还是以前的大小
然后 ext2online -d -v /dev/testvg/testlv 就能使文件系统扩大了，不用 umount

ext3 扩展文件系统(旧版本方法)
除非内核已有 ext2online 补丁，否则在改变 ext2/ext3 文件系统的大小时应卸载它:
umount /dev/testvg/testlv
resize2fs /dev/testvg/testlv
mount /dev/testvg/testlv /home
这里假设 testlv 安装点为/home。在 es2fsprogs-1.19 或以上版本中包含 resize2fs 命令。

reiserfs 文件系统方法
与 ext2 不同，Reiserfs 不必卸载文件系统，如:
resize_reiserfs -f /dev/testvg/testvl
可以在拷贝数据的同时增加减少空间相关命令:lvremove .lvextend,lvreduce .resize2fs resize_reiserfs

xfs 文件系统方法
SGI XFS 文件系统必须在安装的情况下才可改变大小，并且要使用安装点而不是块设备，如:
xfs_growfs /home

114 ☆LVM 缩小 LV

逻辑卷可扩展同样也可缩小，但应在缩小 LV 之前首先减小文件系统，否则将可能导致数据丢失。

umount 要减小的 LV 的步骤
用 ext2resize 减小一个 LV 上的文件系统的大小
用 lvreduce 减小 LV 的大小
之后再 mount，df -Th 看到生效（没有丢失数据）

ext2/ext3
可以使用 LVM 的工具 e2fsadm 操作(旧版本中)，如:
umount /home
e2fsadm -L-1G /dev/testvg/testvl
mount /home
如果采用 resize2fs，就必须知道缩少后卷的块数:
umount /home
resize2fs /dev/testvg/testvl 524288
lvreduce -L-1G /dev/testvg/testvl
mount /home

reiserfs
在缩小 reiserfs 时，应首先卸载它，如:
umount /home
resize_reiserfs -s-1G /dev/testvg/testvl
lvreduce -L-1G /dev/testvg/testvl
mount -treiserfs /dev/testvg/testvl /home

xfs
无法实现。

115 ☆e2fsadm

在 LVM 发行包中有一个称为 e2fsadm 的工具(旧版本)，它同时包含了 lvextend 与 resize2fs 的功能，如:
e2fsadm -L+1G /dev/testvg/testlv
等价于下面两条命令:
lvextend -L+1G /dev/testvg/testlv

resize2fs /dev/testvg/testlv
但用户仍需首先卸载文件系统。

116 ☆LVM 的快照

使用 snapshot 做备份
例如我们要对卷组"test_vg"每晚进行数据库备份，就要采用 snapshot 类型的卷组。这种卷组是其它卷组的一个只读拷贝，它含有在创建 snapshot 卷组时原卷组的所有数据，这意味你可以备份这个卷组而不用担心在备份过程中数据会改变，也不需要暂时关闭数据库卷以备份。

建立 snapshot 卷
一个 snapshot 卷可大可小，但必须有足够的空间存放所有在本 snapshot 卷生存期间改变的数据，一般最大要求是原卷组的 1.1 倍。如空间不够，snapshot 卷将不能使用。
lvcreate -L592M -s -n dbbackup /dev/testvg/testlv　#必须有快照盘"testlv",对哪个生成快照
#针对某一个 lv 的,创建的时候产生快照
lvcreate -- WARNING: the snapshot must be disabled if it gets full
lvcreate -- INFO: using default snapshot chunk size of 64 KB for "/dev/test_vg/dbbackup"
lvcreate -- doing automatic backup of "test_vg"
lvcreate -- logical volume "/dev/test_vg/dbbackup" successfully created

安装 snapshot 卷
现在可以安装该卷:
mkdir /mnt/test_vg/dbbackup
mount /dev/test_vg/dbbackup /mnt/test_vg/dbbackup
mount: block device /dev/test_vg/dbbackup is write-protected, mounting read-only
从上面可以看出，snapshot 卷是只读的。
当使用 XFS 文件系统时，mount 命令要使用 nouuid 与 norecovery 选项:
mount /dev/test_vg/dbbackup /mnt/test_vg/dbbackup -o nouuid,norecovery,ro

备份 snapshot 卷
如采用 tar 向磁带备份:
tar -cf /dev/rmt0 /mnt/test_vg/dbbackup

删除 snapshot 卷
在完成备份后，就可卸载并删除 snapshot 卷。
umount /mnt/test_vg/dbbackup
lvremove /dev/test_vg/dbbackup
lvremove -- do you really want to remove "/dev/test_vg/dbbackup"? [y/n]: y
lvremove -- doing automatic backup of volume group "test_vg"
lvremove -- logical volume "/dev/test_vg/dbbackup" successfully removed

117 ☆LVM 更换卷组硬盘

更换卷组硬盘
由于某种原因，需要用新的硬盘替代卷组中的旧硬盘，如用一 SCSI 硬盘替换 IDE 硬盘，其步骤为:

准备/初始化新硬盘
首先用 pvcreate 命令初始化新的硬盘，如使用整个硬盘:
pvcreate /dev/sdf
pvcreate -- physical volume "/dev/sdf" successfully created

加入卷组
把新硬盘加入卷组:
vgextend test_vg /dev/sdf
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "test_vg"
vgextend -- volume group "test_vg" successfully extended

数据搬家
在移除旧硬盘前，要把其上的数据转移到新硬盘上。在转移数据时，不要求卸载文件系统，但建议在数据转移前进行备份，以防转移进程中意外导致数据丢失。
pvmove 用来实现数据转移，根据数据量的多少，它可能要使用大量的时间，并可降低逻辑卷的性能，因此要在系统不太忙时操作。
pvmove /dev/hdb /dev/sdf
pvmove -- moving physical extents in active volume group "test_vg"
pvmove -- WARNING: moving of active logical volumes may cause data loss!
pvmove -- do you want to continue? [y/n] y
pvmove -- 249 extents of physical volume "/dev/hdb" successfully moved

移除未用硬盘
当数据被转移到其它硬盘后，就可以从卷组中删除这块不再使用的硬盘:
vgreduce dev /dev/hdb
vgreduce -- doing automatic backup of volume group "test_vg"
vgreduce -- volume group "test_vg" successfully reduced by physical volume:
vgreduce -- /dev/hdb
从此，卷组 test_vg 不再使用 IDE 硬盘/dev/hdb，这块硬盘可以从机器中拆下或用作它途。

118 ☆LVM 迁移卷组

把一个卷组转移到其它系统是很容易的(如更换服务器)，这要用命令 vgexport 与 vgimport。

卸载文件系统

为整体搬迁卷组，应首先把它从文件系统中卸载，如:
unmount /mnt/design/users

设置卷组为非活动状态
把卷组从内核中卸载，以避免任何对它可能的操作:
vgchange -a n test_vg
vgchange -- volume group "test_vg" successfully deactivated

Export 卷组
这个操作不是必须的，便它可以防止系统对卷组的访问:
vgexport test_vg
vgexport -- volume group "test_vg" sucessfully exported
当机器关机后，构成卷组的硬盘就可被转移到新的服务器上。

Import 卷组
在新的服务器上，可用 pvscan 查看卷组情况，如在这台计算机上，硬盘新的设备为/dev/sdb，使用 pvscan 可有:
pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- inactive PV "/dev/sdb1" is in EXPORTED VG "test_vg" [996 MB / 996 MB free]
pvscan -- inactive PV "/dev/sdb2" is in EXPORTED VG "test_vg" [996 MB / 244 MB free]
pvscan -- total: 2 [1.95 GB] / in use: 2 [1.95 GB] / in no VG: 0 [0]
现可以 import 卷组 test_vg (同时也激活它)以安装其上的文件系统
vgimport test_vg /dev/sdb1 /dev/sdb2
vgimport -- doing automatic backup of volume group "test_vg"
vgimport -- volume group "test_vg" successfully imported and activated

安装文件系统
mkdir -p /mnt/design/users
mount /dev/test_vg/users /mnt/design/users
在完成以上操作后，原卷组在新的服务器上就可使用了。

119 ☆LVM 分割卷组

这种情况是:需要在系统中加入新的卷组，但没有其它可用新硬盘，而已有的卷组中还有大量空间可用。如向系统加入一个"design"卷组。

检查可用空间
pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- ACTIVE PV "/dev/sda" of VG "dev" [1.95 GB / 0 free]
pvscan -- ACTIVE PV "/dev/sdb" of VG "sales" [1.95 GB / 1.27 GB free]

pvscan -- ACTIVE PV "/dev/sdc" of VG "ops" [1.95 GB / 564 MB free]
pvscan -- ACTIVE PV "/dev/sdd" of VG "dev" [1.95 GB / 0 free]
pvscan -- ACTIVE PV "/dev/sde" of VG "ops" [1.95 GB / 1.9 GB free]
pvscan -- ACTIVE PV "/dev/sdf" of VG "dev" [1.95 GB / 1.33 GB free]
pvscan -- ACTIVE PV "/dev/sdg1" of VG "ops" [996 MB / 432 MB free]
pvscan -- ACTIVE PV "/dev/sdg2" of VG "dev" [996 MB / 632 MB free]
pvscan -- total: 8 [13.67 GB] / in use: 8 [13.67 GB] / in no VG: 0 [0]
我们决定把/dev/sdg1 与/dev/sdg2 分配组 design，但首先要把其上的物理块移到其它卷的空闲空间中(如把卷组 dev 移到/dev/sdf，卷组 ops 移到/dev/sde)。

从选定硬盘移出数据
由于硬盘上的逻辑卷仍在使用，故首先要转移它们的数据。
把所有在使用的物理块从/dev/sdg1 上转移到/dev/sde，及从/dev/sdg2 转移到/dev/sdf。
pvmove /dev/sdg1 /dev/sde
pvmove -- moving physical extents in active volume group "ops"
pvmove -- WARNING: moving of active logical volumes may cause data loss!
pvmove -- do you want to continue? [y/n] y
pvmove -- doing automatic backup of volume group "ops"
pvmove -- 141 extents of physical volume "/dev/sdg1" successfully moved

pvmove /dev/sdg2 /dev/sdf
pvmove -- moving physical extents in active volume group "dev"
pvmove -- WARNING: moving of active logical volumes may cause data loss!
pvmove -- do you want to continue? [y/n] y
pvmove -- doing automatic backup of volume group "dev"
pvmove -- 91 extents of physical volume "/dev/sdg2" successfully moved

创建新卷组
现在把/dev/sdg2 从卷组 dev 从分割出并加入到新卷组 design 中。我们可用 vgreduce 与 vgcreate 完成工作，但 vgsplit 此时更方便:
vgsplit dev design /dev/sdg2
vgsplit -- doing automatic backup of volume group "dev"
vgsplit -- doing automatic backup of volume group "design"
vgsplit -- volume group "dev" successfully split into "dev" and "design"

移除剩余的卷
接下来的工作 把/dev/sdg1 从卷组 ops 中分出并加入卷组 design:
vgreduce ops /dev/sdg1
vgreduce -- doing automatic backup of volume group "ops"
vgreduce -- volume group "ops" successfully reduced by physical volume:
vgreduce -- /dev/sdg1

vgextend design /dev/sdg1
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "design"
vgextend -- volume group "design" successfully extended

建立新逻辑卷及文件系统
在卷组 design 上建立逻辑卷，为今后的方便，现只使用一部分空间:
lvcreate -L750M -n users design
lvcreate -- rounding up size to physical extent boundary "752 MB"
lvcreate -- doing automatic backup of "design"
lvcreate -- logical volume "/dev/design/users" successfully created

mke2fs /dev/design/users
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
96384 inodes, 192512 blocks
9625 blocks (5.00<!--) reserved for the super user
First data block=0
6 block groups
32768 blocks per group, 32768 fragments per group
16064 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

mkdir -p /mnt/design/users
mount /dev/design/users /mnt/design/users/
现在就可使用卷组 design。为方便使用，可把下面一行加入文件/etc/fstab 中:
/dev/design/user /mnt/design/users ext2 defaults 1 2

120 ☆LVM 转变根文件系统为 LVM

注意:强烈要求在进行下面的操作前对系统进行备份，并且把 / 文件系统建立在 LVM 上会导致系统升级很复杂。
在下面的例子中，系统除了/boot 外都安装在同一个分区中，文件系统的情况为:
/dev/hda1 /boot
/dev/hda2 swap

/dev/hda3 /
进行转换的一个必要条件是硬盘上还有足够的空间给分区/dev/hda4 创立 LVM 并把 / 分区的内容都复制到 LVM 上，否则:
1. / 分区还有至少一半空间空闲，可以缩减 / 分区，并把分出的空间划分到分区/dev/hda4;
2.硬盘上已无足够空间，必须使用第二块硬盘，如/dev/hdb。
在完成以上准备及备份系统后，可继续以下步骤:
1.确认使用的 Linux 内核支持 LVM，并且在编译时设置了 CONFIG BLK DEV RAM 与 CONFIG BLK DEV INITRD 。
2.设置/dev/hda4 分区类型为 LVM(8e):
fdisk /dev/hda
Command (m for help): t
Partition number (1-4): 4
Hex code (type L to list codes): 8e
Changed system type of partition 4 to 8e (Unknown)
Command (m for help): w
3.设置 LVM:
"初始化 LVM (vgscan)
vgscan
"转变分区为 PV:
pvcreate /dev/hda4
"建立卷组:
vgcreate vg /dev/hda4
"建立逻辑卷用以存放根系统:(这里假设空间为 250MB)
lvcreate -L250M root vg
4.在逻辑卷上建立文件系统并把系统复制到其上:
mke2fs /dev/vg/root
mount /dev/vg/root /mnt/
find / -xdev | cpio -pvmd /mnt
5.修改新系统的 fstab 文件/mnt/etc/fstab，使 / 安装到/dev/vg/root:
/dev/hda3 / ext2 defaults 1 1
改变为:
/dev/vg/root / ext2 defaults 1 1
6.创建 LVM 初始化 RAM 盘:
lvmcreate_initrd
此处要确认为 lvmcreate_init 给出正确的 initrd image 文件名，它应在/boot/ 目录下。
7. 在/etc/lilo.conf 中为 LVM 加入新入口项，其形式如下:
image = /boot/KERNEL_IMAGE_NAME
label = lvm
root = /dev/vg/root
initrd = /boot/INITRD_IMAGE_NAME
ramdisk = 8192

此处 KERNEL IMAGE NAME 是支持 LVM 的内核，INITRD IMAGE NAME 指由 lvmcreate_initrd 建立的 initrd image。如果 LVM 的配置很多，可以把 ramdisk 设置的大一些:此处为 8192，缺省为 4096。在 lvmcrate_initrd 的输出中有如下一行:
lvmcreate_initrd -- making loopback file (6189 kB)
其中括号中的数值为实际所需大小。
8.运行 LILO，设置 BOOT 扇区:
lilo
9.重启计算机，在 LILO 提示符处输入"lvm" 启动计算机，此时系统的根文件系统是新建立的逻辑卷。此后可在 LILO 配置文件/etc/lilo.conf 中加入以下一行:
default=lvm
并运行 lilo 设置缺省启动项为 lvm。
如果系统未能正常启动，可能的原因是内核不支持 LVM、initrd image 不正确等等。
10.在正常启动后，就可把硬盘其它分区:/dev/hda3 加入 LVM。
"首先设置分区类型为 8e(LVM)
fdisk /dev/hda
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 8e
Changed system type of partition 3 to 8e (Unknown)
Command (m for help): w
"把它初始化为 PV，并加入卷组中:
pvcreate /dev/hda3
vgextend vg /dev/hda3

121 ☆LVM 共享 LVM 卷

LVM 不支持物理共享访问，这会导致数据的丢失。
在使用 fibre-channel 或 shared-SCSI 的环境中，多台计算机以物理方式直接访问一组硬盘，于是可以使用 LVM 把这些硬盘分为不同的逻辑卷。如果需要共享数据，则应使用 GFS。

122 ☆LVM 系统启动/关闭

"为使系统启动时可自动激活并使用 LVM，可将以下几行添加到启动 rc 脚本中:
/sbin/vgscan
/sbin/vgchange -a y
这些行将浏览所有可用的卷组并激活它们。要注意的是，它们应在安装卷组上的文件系统操作之前被执行，否则将无法正常安装文件系统。
"在系统关机时，要关闭 LVM，这可将以下这行添加到关机 rc 脚本中，并确保它在卸装了所有文件系统后执行:
/sbin/vgchange -a n

123 ☆LVM 磁盘分配策略

一个磁盘上的多个分区
LVM 允许 PV 建立在几乎所有块设备上，如整个硬盘、硬盘分区、Soft RAID:
pvcreate /dev/sda1
pvcreate /dev/sdf
pvcreate /dev/hda8
pvcreate /dev/hda6
pvcreate /dev/md1
所以在一块硬盘上可以有多个 PV / 分区，但一般建议一块硬盘上只有一个 PV:
"便于管理，易于处理错误
"避免交错方式中性能下降。LVM 不能辨别两个 PV 是否在同一硬盘上，故当采用交错方式时，会导致性能更差。
但在某些情况下可采用:
"把已存在的系统合并到 LVM 中。在一个只有少数硬盘的系统中，转换为 LVM 时需在在各分区之间转移数据。
"把一个大硬盘分给不同的 VG 使用。
当一个 VG 的有不同的 PV 在同一硬盘时，创建交错方式的 LV 时应注意使用哪一个 PV。

Sun disk labels
仅在 SUN 的 SPARC 系统中有此问题。

124 ☆lvm 实验

安装需求(requirements)：检查系统是否安了 lvm 的 module
1. lsmod | grep -i lvm
2. modprobe lvm-mod
3.apt-get install lvm* lvm-common evms-lvmutils
或者
rpm -q lvm , rpm -ivh lvm-xxxx.rpm with redhat

准备虚拟硬盘 prepare a fake disk
感谢 linux lvm 支持 loopback (loop device)，才让我们有机会建立 lvm 下的虚拟硬盘，就跟真硬盘一样用
1. dd if=/dev/zero of=lvm0.iso count=5000
2. dd if=/dev/zero of=lvm1.iso count=5000
3. dd if=/dev/zero of=lvm2.iso count=5000
4. dd if=/dev/zero of=lvm3.iso count=5000

5.losetup /dev/loop0 lvm0.iso
6.losetup /dev/loop1 lvm1.iso
7.losetup /dev/loop2 lvm2.iso

8.losetup /dev/loop3 lvm3.iso

好了，我们这里建立了可用的虚拟硬盘了，要删除这样的硬盘：
losetup -d /dev/loopX

下面我们建立几个物理卷(？) Physical Volume (对不起，我不知中文是怎么翻译的)
1. vgscan
2. pvcreate /dev/loop0
3. pvcreate /dev/loop1
4. pvcreate /dev/loop2
5. pvscan <--- 非常有用
6. lvmdiskscan <-- 查看所有硬盘(包含虚拟)的情况
7。pvdisplay /dev/loop0 <--看看

下面我们建立一个 volume group(对不起，我不知中文是怎么翻译的),就是这个概念让我们可以随时加大分区。-s 是叫 PE (Physical Extend) 的大小是 8K，缺省值是 4Mo,PE 是个基础块，它的大小直接影响物理卷的大小，因为 linux 核最大限制是 65536 个 PE，PE 最小值是 8K，最大值是 16G。理论最大值是 16Gx65536PE=1 petabytes.大家可以看第四张图，可以帮助理解。

下面命令的意思是建立一个叫 vg01 的 volume group，并把硬盘 loop0，loop1 加到入其中:
vgcreate -s 8k vg01 /dev/loop0 /dev/loop1

如果 vg01 不够用了，我们还可以用下面方法加入其它硬盘：
vgextend vg01 /dev/loop2

在 volume group vg01 下建立逻辑卷(Logicial volume creation)，名字叫 lv01，大小为 4M:
1. lvcreate -L4M -n lv01 vg01

再看一眼 vg01:
2. vgdisplay vg01

看看新建的逻辑卷 lv01:
3. lvdisplay /dev/vg01/lv01

也可以换个名字：
4。lvrename /dev/vg01/lv01 /dev/vg01/lv_test

给逻辑卷进行格式化，我用 ext3，你也可以用 vfat,reiserfs,ext2,不过实际应用应该注意到如果你想在 windows 下共享这个 vfat 的话，windows 不会认为是一个分区，也不认识它的格式，这是因为我们这里利用了 linux lvm 支持 loopback 的特性，所以看不出来，实际上如果你真有几张硬盘的话，想用 lvm，也必须先把分区定为 linux lvm type，然后再格式化成 vfat 。因为 windows 不认识 lvm,所以也不会认识你的 vfat 了。我已经做过实验了！大家可以不再化时间了。

1。mkfs -t ext3 -j /dev/vg01/lv_test
2。mount /dev/vg01/lv_test /mnt

可以用了！

下面简单给出删除/扩充/减小 volume groupe 的命令，还有所有有关 lvm 清单，请大家小心，下面的命令适应与 ext2/ext3, 其它如 reiserfs,jfs,xfs，命令不一样。有感兴趣的朋友可以给我发信，我再详细给出命令。

delete volume groupe
1。vgchange -a n /dev/vg01
2。vgremove /dev/vg01

extend
1。lvextend -L+1G /dev/vg01/lv01
2。umount /mountingpoint
3。resize2fs /dev/vg01/lv01
4。mount /dev/vg01/lv01 /mountingpoint

reduce:
1。umount /mountingpoint
2。resize2fs /dev/vg01/lv01
3。lvreduce -L-1G /dev/vg01/lv01
4。mount /dev/vg01/lv01 /mountingpoint

125☆mdadm

mdadm 是一个全新的 raid 管理工具,raidtools 需要维护配置文件/etc/raidtab 太麻烦了!
mdadm 已经不使用 /etc/raidtab 了
mdadm 是 linux 下用于管理软件 raid 的工具

基本语法 ：
mdadm [mode] [options]

[mode] 有 7 种：

Assemble：将以前定义的某个阵列加入当前在用阵列。
Build：Build a legacy array ，每个 device 没有 superblocks
Create：创建一个新的阵列，每个 device 具有 superblocks
Manage： 管理阵列，比如 add 或 remove
Misc：允许单独对阵列中的某个 device 做操作，比如抹去 superblocks 或 终止在用的阵列。
Follow or Monitor:监控 raid 1,4,5,6 和 multipath 的状态
Grow：改变 raid 容量或 阵列中的 device 数目

可用的 [options]:
-A, --assemble：加入一个以前定义的阵列
-B, --build：Build a legacy array without superblocks.
-C, --create：创建一个新的阵列
-Q, --query：查看一个 device，判断它为一个 md device 或是 一个 md 阵列的一部分
-D, --detail：打印一个或多个 md device 的详细信息
-E, --examine：打印 device 上的 md superblock 的内容
-F, --follow, --monitor：选择 Monitor 模式
-G, --grow：改变在用阵列的大小或形态
-h, --help：帮助信息，用在以上选项后，则显示该选项信息
--help-options:Display more detailed help about command line parsing and some commonly used options.
-V, --version:Print version information for mdadm.
-v, --verbose：显示细节
-b, --brief：较少的细节。用于 --detail 和 --examine 选项
-f, --force:Be more forceful about certain operations. See the various modes of the exact meaning of this option in different contexts.
-c, --config= ：指定配置文件，缺省为 /etc/mdadm/mdadm.conf
Specify the config file. Default is /etc/mdadm.conf. If the config file given is partitions then nothing will be read, but mdadm will act as though the config file contained exactly DEVICE partitions and will read /proc/partitions to find a list of devices to scan. If the word none is given for the config file, then mdadm will act as though the config file were empty.
-s, --scan：扫描配置文件或 /proc/mdstat 以搜寻丢失的信息。配置文件/etc/mdadm/mdadm.conf
scan config file or /proc/mdstat for missing information. In general, this option gives mdadm permission to get any missing information, like component devices,
array devices, array identities, and alert destination from the configuration file: /etc/mdadm.conf. One exception is MISC mode when using --detail or --stop in which case --scan says to get a list of array devices from /proc/mdstat.

create 或 build 使用的选项:
-c, --chunk=:Specify chunk size of kibibytes. 缺省为 64.
--rounding=:Specify rounding factor for linear array (==chunk size)

-l, --level=:设定 raid level.

Set raid level. When used with --create, options are: linear, raid0, 0, stripe, raid1, 1, mirror, raid4, 4, raid5, 5, raid6, 6, multipath, mp. Obviously some of these are synonymous.

When used with --build, only linear, raid0, 0, stripe are valid.

--create 可用:linear, raid0, 0, stripe, raid1,1, mirror, raid4, 4, raid5, 5, raid6, 6, multipath, mp.

--build 可用：linear, raid0, 0, stripe.

-p, --parity=：设定 raid5 的奇偶校验规则:eft-asymmetric, left-symmetric, right-asymmetric, right-symmetric, la, ra, ls, rs.缺省为 left-symmetric
parity-algorithm 表示 raid5 的奇偶效验的运算法则，可用选择有：
left-symmetric left-asymmetric right-symmetric right-asymmetric
最佳性能的是：left-symmetric

--layout=:类似于--parity

-n, --raid-devices=:指定阵列中可用 device 数目，这个数目只能由 --grow 修改

Specify the number of active devices in the array. This, plus the number of spare devices (see below) must equal the number of component-devices (including "missing" devices) that are listed on the command line for --create. Setting a value of 1 is probably a mistake and so requires that --force be specified first. A value of 1 will then be allowed for linear, multipath, raid0 and raid1. It is never allowed for raid4 or raid5.

This number can only be changed using --grow for RAID1 arrays, and only on kernels which provide necessary support.

-x, --spare-devices=：指定初始阵列的富余 device 数目

Specify the number of spare (eXtra) devices in the initial array. Spares can also be added and removed later. The number of component devices listed on the command

line must equal the number of raid devices plus the number of spare devices.

-z, --size=：组建 RAID1/4/5/6 后从每个 device 获取的空间总数

Amount (in Kibibytes) of space to use from each drive in RAID1/4/5/6. This must be a multiple of the chunk size, and must leave about 128Kb of space at the end of the drive for the RAID superblock. If this is not specified (as it normally is not) the smallest drive (or partition) sets the size, though if there is a variance among the drives of greater than 1%, a warning is issued.

This value can be set with --grow for RAID level 1/4/5/6. If the array was created with a size smaller than the currently active drives, the extra space can be

accessed using --grow.

--assume-clean:目前仅用于 --build 选项

Tell mdadm that the array pre-existed and is known to be clean. This is only really useful for Building RAID1 array. Only use this if you really know what you are doing. This is currently only supported for --build.

-R, --run:阵列中的某一部分出现在其他阵列或文件系统中时，mdadm 会确认该阵列。此选项将不作确认。

Insist that mdadm run the array, even if some of the components appear to be active in another array or filesystem. Normally mdadm will ask for confirmation before including such components in an array. This option causes that question to be suppressed.

-f, --force:通常 mdadm 不允许只用一个 device 创建阵列，而且创建 raid5 时会使用一个 device 作为 missing drive。此选项正相反。

Insist that mdadm accept the geometry and layout specified without question. Normally mdadm will not allow creation of an array with only one device, and will try to create a raid5 array with one missing drive (as this makes the initial resync work faster). With --force, mdadm will not try to be so clever.

-a, --auto{=no,yes,md,mdp,part,p}{NN}：

Instruct mdadm to create the device file if needed, and to allocate an unused minor number. "yes" or "md" causes a non-partitionable array to be used. "mdp", "part" or "p" causes a partitionable array (2.6 and later) to be used. The argumentment can also come immediately after "-a". e.g. "-ap".

For partitionable arrays, mdadm will create the device file for the whole array and for the first 4 partitions. A different number of partitions can be specified at the end of this option (e.g. --auto=p7). If the device name ends with a digit, the partition names add an underscore, a ’p’, and a number, e.g. "/dev/home1_p3". If there is no trailing digit, then the partition names just have a number added, e.g. "/dev/scratch3".

For assemble:

-u, --uuid=

uuid of array to assemble. Devices which don’t have this

126 ☆建立 RAID

可以在磁盘级别或者分区级别上做

磁盘级别的时候,不用指定类型

创建 4 个新 raid 分区,过程如下：改为 fd 类型.即(Linux raid autodetect)

[root@localhost ~]# fdisk /dev/sdc

使用 mdadm 命令来建立一个 RAID 0 阵列/dev/md0.

```
[root@localhost ~]# mdadm -C /dev/md0 --level=0 --raid-devices=4 /dev/sdc1 /dev/sdc2 /dev/sdc3
/dev/sdc4
mdadm: /dev/sdc1 appears to contain an ext2fs file system
       size=104128K  mtime=Thu Jan  1 08:00:00 1970
mdadm: /dev/sdc1 appears to be part of a raid array:
       level=0 devices=4 ctime=Wed Aug  2 04:07:42 2006
mdadm: /dev/sdc2 appears to be part of a raid array:
       level=0 devices=4 ctime=Wed Aug  2 04:07:42 2006
mdadm: /dev/sdc3 appears to be part of a raid array:
       level=0 devices=4 ctime=Wed Aug  2 04:07:42 2006
mdadm: /dev/sdc4 appears to be part of a raid array:
       level=0 devices=4 ctime=Wed Aug  2 04:07:42 2006
Continue creating array? y
mdadm: array /dev/md0 started.
```

mdadm -C /dev/md0 -l5 -n4 /dev/sdc[1234] 这里就在三个分区上创建了 raid5 软件陈列, 新的分区设备号为 md0

127 ☆查看 RAID

cat /proc/mdstat

```
[root@localhost ~]# mdadm -D /dev/md0
/dev/md0:
           Version : 00.90.01
     Creation Time : Wed Aug  2 04:07:42 2006
        Raid Level : raid0
        Array Size : 104128 (101.69 MiB 106.63 MB)
      Raid Devices : 4
     Total Devices : 4
Preferred Minor : 0
       Persistence : Superblock is persistent

       Update Time : Wed Aug  2 04:07:43 2006
             State : clean
    Active Devices : 4
Working Devices : 4
    Failed Devices : 0
     Spare Devices : 0
```

Chunk Size : 64K

Number	Major	Minor	RaidDevice	State	
0	8	33	0	active sync	/dev/sdc1
1	8	34	1	active sync	/dev/sdc2
2	8	35	2	active sync	/dev/sdc3
3	8	36	3	active sync	/dev/sdc4

UUID : dd7bc5b8:694f0193:4056ea93:f28049dc
Events : 0.1

128 ☆挂载 RAID

在新建的 raid 上创建 ext3
[root@localhost ~]# mkfs.ext3 /dev/md0
mke2fs 1.35 (28-Feb-2004)
max_blocks 805961728, rsv_groups = 24596, rsv_gdb = 192
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
393600 inodes, 787072 blocks
39353 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=809500672
25 block groups
32768 blocks per group, 32768 fragments per group
15744 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912

Writing inode tables: done
inode.i_blocks = 9224, i_size = 4243456
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.

挂载 raid 阵列
[root@localhost ~]# mkdir /data
[root@localhost ~]# mount /dev/md0 /data
[root@localhost ~]# cd /data

[root@localhost data]# ls
lost+found
[root@localhost data]# df -H /data
Filesystem 容量 已用 可用 已用% 挂载点
/dev/md0 3.2G 39M 3.0G 2% /data
[root@localhost data]# cat /proc/mdstat
Personalities : [raid0]
md0 : active raid0 sdc4[3] sdc3[2] sdc2[1] sdc1[0]
104128 blocks 64k chunks

unused devices: <none>
[root@localhost ~]#umount /data

129 ☆RAID 设置热备盘

mdadm -C /dev/md0 -l5 --raid-devices=3 /dev/sdc1 /dev/sdc2 /dev/sdc3 --spare-device=1 /dev/sdc4
/dev/sda4 就是 spare devcie
或
mdadm -C /dev/md0 -l5 --n3 /dev/sdc1 /dev/sdc2 /dev/sdc3 -x1 /dev/sdc4

Number Major Minor RaidDevice State
0 8 7 0 active sync /dev/sda7
1 8 8 1 active sync /dev/sda8
2 8 9 2 active sync /dev/sda9
3 8 10 -1 spare /dev/sda10

为了让系统重新启动后自动挂载,可以修改一下/etc/fstab 文件,添加一行!
/dev/md0 /raid5disk auto defaults 0 0

这样系统重新启动后会自动将/dev/md0 挂接到 /raid5disk 目录下!

130 ☆RAID 阵列转移

先停止活动的磁盘阵列:
mdadm -S /dev/md0

再装配到/dev/md1:
[root@localhost data]# mdadm -A /dev/md1 /dev/sdc1 /dev/sdc2 /dev/sdc3 /dev/sdc4
mdadm: /dev/md1 has been started with 4 drives.

[root@localhost data]# mdadm -D /dev/md1
/dev/md1:

Version : 00.90.01
Creation Time : Sat Nov 19 10:01:47 2005
Raid Level : raid0
Array Size : 3148288 (3.00 GiB 3.22 GB)
Raid Devices : 4
Total Devices : 4
Preferred Minor : 1
Persistence : Superblock is persistent

Update Time : Sat Nov 19 10:01:47 2005
State : clean
Active Devices : 4
Working Devices : 4
Failed Devices : 0
Spare Devices : 0

Chunk Size : 64K

Number Major Minor RaidDevice State
0 3 9 0 active sync /dev/hda9
1 3 10 1 active sync /dev/hda10
2 3 11 2 active sync /dev/hda11
3 3 12 3 active sync /dev/hda12
UUID : da00d68c:ec3e8057:dafaf27a:fb0c2fa3
Events : 0.1

---

131 ☆删除 RAID

mdadm -S /dev/md0

rm /dev/md0(不必要)

del /etc/mdadm.conf

del raid mount in /etc/fstab and autofs

del raid part use fdisk or parted

132 ☆RAID 替换磁盘(在线方式)

mdadm /dev/md0 -a /dev/sdb6
先前如果没有加热备盘的话

mdadm /dev/md0 -f /dev/sdc1
设置错误

mdadm /dev/md0 -r /dev/sdc1
移除/dev/sdc1

133 ☆mdadm.conf

建立 mdadm.conf 这步可省略，但为了便于以后管理，建议使用
[root@localhost ~]#echo DEVICE /dev/hd*[0-9] /dev/sd*[0-9] > /etc/mdadm.conf
[root@localhost ~]#mdadm --detail --scan >> /etc/mdadm.conf

134 ☆mkraid

mkraid -R /dev/md2
mkfs -j /dev/md2
格式化设备

135 ☆dd

功能:把指定的输入文件拷贝到指定的输出文件中，并且在拷贝过程中可以进行格式转换。可以用该命令实现 DOS 下的 diskcopy 命令的作用。先用 dd 命令把软盘上的数据写成硬盘的一个寄存文件，再把这个寄存文件写入第二张软盘上，完成 diskcopy 的功能。需要注意的是，应该将硬盘上的寄存文件用 rm 命令删除掉。系统默认使用标准输入文件和标准输出文件。

语法:dd [选项]
if =输入文件(或设备名称)。
of =输出文件(或设备名称)。
bs = bytes 同时设置读/写缓冲区的字节数(等于设置 ibs 和 obs)。
cbs = byte 一次转换 bytes 字节。
ibs = bytes 一次读取 bytes 字节，即读入缓冲区的字节数。
obs = bytes 一次写入 bytes 字节，即写入缓冲区的字节数。
skip = blocks 跳过读入输入缓冲区开头的 ibs*blocks 块。
seek = blocks 跳过读入输出缓冲区开头的 obs*blocks 块
count=blocks 只拷贝输入的 blocks 块。
conv = ASCII 把 EBCDIC 码转换为 ASCIl 码。
conv = ebcdic 把 ASCIl 码转换为 EBCDIC 码。

conv = ibm 把 ASCIl 码转换为 alternate EBCDIC 码。
conv = block 把变动位转换成固定字符。
conv = ublock 把固定位转换成变动位。
conv = ucase 把字母由小写转换为大写。
conv = lcase 把字母由大写转换为小写。
conv = notrunc 不截短输出文件。
conv = swab 交换每一对输入字节。
conv = noerror 出错时不停止处理。
conv = sync 把每个输入记录的大小都调到 ibs 的大小(用 NUL 填充)。

BLOCKS 和 BYTES 可已有如下后缀:
xM M
c 1
w 2
b 512
kB 1000
K 1024
MB 1000*1000
M 1024*1024
GB 1000*1000*1000
G 1024*1024*1024,
以及 T, P, E, Z, Y 等等

将文件 sourcefile 拷贝到文件 destfile
dd if=sourcefile of=destfile

要把一张软盘的内容拷贝到另一张软盘上，利用/tmp 作为临时存储区。把源盘插入驱动器中，输入下述命令:
$ dd if =/dev/fd0 of = /tmp/tmpfile

拷贝完成后，将源盘从驱动器中取出，把目标盘插入，输入命令:
$ dd if = /tmp/tmpfile of =/dev/fd0

软盘拷贝完成后，应该将临时文件删除:
$ rm /tmp/tmpfile

把 net.i 这个文件写入软盘中，并设定读/写缓冲区的数目。
(注意:软盘中的内容会被完全覆盖掉)
$ dd if = net.i of = /dev/fd0 bs = 16384

例 3:将文件 sfile 拷贝到文件 dfile 中。
$ dd if=sfile of=dfile

dd 命令去备份文件指定的部分了。

举例说明：假如我有一个文件 abc.gz，大小为 83456k，我想用 dd 命令实现如下备份结果：首先将备份分成三个部分，第一部分为备份文件 abc.gz 的前 10000k，第二部分为中间的 70000k，最后备份后面的 3456k.

备份方法如下三条命令：
dd if=abc.gz of=abc.gz.bak1 bs=10000k count=1
dd if=abc.gz of=abc.gz.bak2 bs=10000k skip=1 count=7
dd if=abc.gz of=abc.gz.bak3 bs=10000k skip=8

恢复方法如下：
dd if=abc.gz.bak1 of=abc.gz
dd if=abc.gz.bak2 of=abc.gz bs=10000k seek=1
dd if=abc.gz.bak3 of=abc.gz bs=10000k seek=8

这时你查看一下恢复的文件将和你原来的文件一模一样，说明备份成功!

说明一下:
bs=xxx 这个选项是指你要备份时一次性创建的块大小,而 count=xxx 则是指从备份的开头开始算总共备份多少块.
也就是说假如你的 bs=3k 而 count=5，则说明你一个块为 3k，总共 5 个块，则备份了原文件的 15k(3k*5=15k).
而 skip=xxx 则是在备份时对 if 后面的部分也就是原文件跳过多少块再开始备份，相反 seek=xxx 则是在备份时对 of 后面的部分也就是目标文件跳过多少块再开始写。

136 ☆fdformat

使用权限: 所有使用者

使用说明 :
对指定的软盘装置进行低阶格式化。

使用方式:
fdformat [-n] device

参数:
-n 软盘格式化后不作检验。关闭确认功能。这个选项会关闭格式化之后的确认步骤。

device 指定要进行格式化的设备，通常是下述设备之一:
/dev/fd0d360

/dev/fd0h1200
/dev/fd0D360
/dev/fd0H360
/dev/fd0D720
/dev/fd0H720
/dev/fd0h360
/dev/fd0h720
/dev/fd0H1440

如果使用像是 /dev/fd0 之类的装置，如果里面的磁碟不是标准容量，格式化可能会失败。在这种情况之下，使用者可以用 setfdprm 指令先行指定必要参数。

范例:
fdformat -n /dev/fd0h1440
将软盘格式化成 1.4MB 的磁片。并且省略确认的步骤。
然后 mkfs

137 ☆mformat

使用权限: 所有使用者

使用方式:
mformat [-t cylinders] [-h heads] [-s sectors] [-l volume_label] [-F] [-I fsVer-sion] [-S sizecode] [-2 sectors_on_track_0] [-M software_sector_size] [-a] [-X] [-C] [-H hidden_sectors] [-r root_sectors] [-B boot_sector] [-0 rate_on_track_0] [-A rate_on_other_tracks] [-1] [-k] drive:

在已经做过低阶格式化的磁片上建立 DOS 文件系统。如果在编译 mtools 的时候把 USE_2M 的参数打开，部分与 2M 格式相关的参数就会发生作用。否则这些参数(像是 S,2,1,M)不会发生作用。

参数:

-t 磁柱(synlider)数
-h 磁头(head)数
-s 每一磁轨的磁区数
-l 标签
-F 将磁碟格式化为 FAT32 格式，不过这个参数还在实验中。
-I 设定 FAT32 中的版本号。这当然也还在实验中。
-S 磁区大小代码，计算方式为 sector = 2^(大小代码+7)
-c 磁丛(cluster)的磁区数。如果所给定的数字会导致磁丛数超过 FAT 表的限制，mformat 会自动放大磁区数。
-s

-M 软体磁区大小。这个数字就是系统回报的磁区大小。通常是和实际的大小相同。
-a 如果加上这个参数，mformat 会产生一组 Atari 系统的序号给这块软盘。
-X 将软盘格式化成 XDF 格式。使用前必须先用 xdfcopy 指令对软盘作低阶格式化的动作。
-C 产生一个可以安装 MS-DOS 文件系统的磁碟影像档(disk image)。当然对一个实体磁碟机下这个参数是没有意义的。
-H 隐藏磁区的数目。这通常适用在格式化硬碟的分割区时，因为通常一个分割区的前面还有分割表。这个参数未经测试，能不用就不用。
-n 磁碟序号
-r 根目录的大小，单位是磁区数。这个参数只对 FAT12 和 FAT16 有效。
-B 使用所指定的文件或是设备的开机磁区做为这片磁片或分割区的开机磁区。当然当中的硬体参数会随之更动。
-k 尽量保持原有的开机磁区。
-0 第 0 轨的资料传输率
-A 第 0 轨以外的资料传输率
-2 使用 2m 格式
-1 不使用 2m 格式

范例:
mformat a:
这样会用预设值把 a: (就是 /dev/fd0)里的磁碟片格式化。

138 ☆mkdosfs

使用权限: 所有使用者

使用方式: mkdosfs [-c | -l filename]
[-f number_of_FATs]
[-F FAT_size]
[-i volume_id]
[-m message_file]
[-n volume_name]
[-r root_dir_entry]
[-s sector_per_cluster]
[-v]
device
[block_count]

说明: 建立 DOS 文件系统。 device 指你想要建立 DOS 文件系统的装置代号。像是 /dev/hda1 等等。 block_count 则是你希望配置的区块数。如果 block_count 没有指定则系统会自动替你计算符合该装置大小的区块数。

参数:

-c 建立文件系统之前先检查是否有坏轨。
-l 从得定的文件中读取坏轨记录。
-f 指定文件配置表(FAT , File Allocation Table)的数量。预设值为 2 。目前 Linux 的 FAT 文件系统不支援超过 2 个 FAT 表。通常这个不需要改。
-F 指定 FAT 表的大小，通常是 12 或是 16 个位元组。12 位元组通常用于磁碟片，16 位元组用于一般硬碟的分割区，也就是所谓的 FAT16 格式。这个值通常系统会自己选定适当的值。在磁碟片上用 FAT16 通常不会发生作用，反之在硬碟上用 FAT12 亦然。
-i 指定 Volume ID。一般是一个 4 个位元组的数字，像是 2e203a47 。如果不给系统会自己产生。
-m 当使用者试图用这片磁片或是分割区开机，而上面没有作业系统时，系统会给使用者一段警告讯息。这个参数就是用来变更这个讯息的。你可以先用文件编辑好，然后用这个参数指定，或是用
-m -
这样系统会要求你直接输入这段文字。要特别注意的是，文件里的字串长度不要超过 418 个字，包括展开的跳栏符号(TAB)和换行符号(换行符号在 DOS 底下算两个字元！)
-n 指定 Volume Name，就是磁碟标签。如同在 DOS 底下的 format 指令一样，给不给都可以。没有预设值。
-r 指定根目录底下的最大文件数。这里所谓的文件数包括目录。预设值是在软盘上是 112 或是 224 ，在硬碟上是 512。没事不要改这个数字。
-s 每一个磁丛(cluster)的磁区数。必须是 2 的次方数。不过除非你知道你在作什么，这个值不要乱给。
-v 提供额外的讯息

范例:
mkdosfs -n Tester /dev/fd0 将 A 软盘格式化为 DOS 格式，并将标签设为 Tester

139 ☆badblocks

$ badblocks /dev/fd0H1440 1440 > bad-blocks
把坏块输出到一个文件

$ fsck -t ext2 -l bad-blocks /dev/fd0H1440
利用坏块文件来检查磁盘

140 ☆autofs

mount 是用来挂载文件系统的，可以在启动的时候挂载也可以在启动后挂载.对于本地固定设备，如硬盘可以使用 mount 挂载，而光盘，软盘，NFS，SMB 等文件系统具有动态性，即需要的时候才有必要挂载，光驱和软盘我们一般知道什么时候需要挂载，但 NFS，SMB 共享等就不一定知道了，即我们一般不能及时知道 NFS 共享和 SMB 什么时候可以挂载，而 autofs 服务就提供这种功能，好像 windows 中的光驱自动打开功能，能够及时挂载动态加载的文件系统。免去我们手动挂在麻烦。

要实现光驱，软盘等的动态自动挂载，需要进行相关的配置。

autofs 的两个配置文件在/etc/auto.master 和/etc/auto.misc

141 ☆auto.master

主要配置文件(/etc/auto.master)
/etc/auto.master 定义了 mount 目录和 mount 所需的配置文件名及其空闲时自动 umount 的时间。
修改/etc/auto.master，设置挂载点

格式：挂载集群点　配置文件

举例：
/mnt　　/etc/auto.misc --timeout 60 (/etc/auto.misc 中配置挂载项挂载在/mnt 下,如果 60 秒内没有活动就自动卸载.)
/mnt/net　/etc/auto.net　　(/etc/auto.net 中配置挂载项挂载在/mnt/net 下)

142 ☆auto.misc

配置文件的设置(/etc/auto.misc)

配置文件用来设置需要挂载的文件系统，每行为一个文件系统，如果一行写不完，可以用\换行，格式如下：
相对挂载点 挂载参数　文件系统位置

各种文件系统的挂载实例如下(这里以/etc/auto.misc 为例)
nfs　-ro,soft,intr　　172.16.0.3:/pub/syd168　(可以使用域名)
cd　-fstype=iso9660,iocharset=cp936,ro　:/dev/cdrom
fd　-fstype=vfat　　:/dev/fd0
win　-fstype=smbfs　　://sambaserver/syd168
local　-fstype=ext3　　:/dev/hda1
smb　-fstype=smbfs　　://192.168.152.130/syd1
rpms　　-fstype=nfs　　192.168.152.130:/mnt/rpms

!!!!!!!!!!!!nfs 和 samba 得格式不一样的

说明：
以上的挂载分别挂载的是 nfs,cdrom,floppy,windows 共享，本地文分区。挂载成功后，访问的位置分别是：/mnt/nfs,/mnt/cd,/mnt/fd,/mnt/win,/mnt/local。

对于包含帐户密码的 smb 的挂载为

smb -fstype=smbfs ,username=user,password=pass ://192.168.152.130/syd5

143 ☆启动 autofs

默认 autofs 是启动的
#service autofs start

144 ☆测试 autofs

访问挂载文件系统的方法
#cd /misc/相对挂载点

挂载文件系统的卸载
#umount /misc/相对挂载点

145 ☆Linux 配额

实现用户磁盘使用量的限制就是所谓的磁盘配额（quota）

Linux 中的磁盘配额分为用户配额和组的配额，组的配额是对某组中全体用户的综合限制。

对磁盘配额的限制一般是从一个用户占用磁盘大小和所有文件的数量两个方面来进行的。

Linux 中的磁盘配额按是否可有一定的超越又分为软限制（可以超越）和硬性限制（禁止超越），但软限制超过期限后自动变为硬性限制

Linux 中的磁盘配额按限制的项目不同，可分为空间限制(blocks 大小)和文件数限制(inodes 个数)。

软限制：一个用户在文件系统可拥有的最大磁盘空间和最多文件数量，在某个宽限期内可以暂时超过这个限制。

硬限制：一个用户可拥有的磁盘空间或文件的绝对数量，绝对不允许超过这个限制。

146 ☆QUOTA 使用前配置

先看内核是否支持
[root@panda redhat]# grep CONFIG_QUOTA /boot/config-2.6.9-5.EL
CONFIG_QUOTA=y
CONFIG_QUOTACTL=y
就代表是支持的
make menuconfig or make xconfig 来更新核心

147 ☆quotacheck

初始化限额数据库,扫描系统中的限额
quotacheck -cvuga
或者
quotacheck -cvug /dev/sdb1(在某个分区上)
检查并创建磁盘配额数据库文件
c 创建
v 显示信息
u 用户
g 组
a 所有激活限额的磁盘分区
m 重新挂载扫描过的文件系统

每个分区的根目录下面有配额数据库文件
aquota.group
aquote.user

要写入 fstab,重启才生效或者 quotaon 来激活
mount -o remount
或者直接指定
-o usrquota,grpquota

0 4 * * 6 /sbin/quotacheck -avug
定时执行

148 ☆quotaon

/sbin/quotaon
激活限额

quotaon 设备名|目录名

quotaon -a
直接激活所有

149 ☆quotaoff

/sbin/quotaoff
关闭限额

quotaoff 挂载点|设备
关闭磁盘配额

150☆edquota

指定用户或组的分配磁盘空间和节点数量
-u 用户名
-g 组名

/etc/mtab
磁盘配额配置文件

edquota -u panda
编辑指定用户

-p 参数（prototype）可以对已有的用户设置进行拷贝。
edquota -p panda -u Jack Tom Chen
拷贝 panda 到其他用户

edquota -g panda
编辑组的限额

配额格式说明：
filesystem blocks soft hardinodes soft hard-------自动添加的行，不能删除！

filesystem-正在设置的文件系统，不要修改或删除！
block-当前已经使用的磁盘空间,块个数，块的大小为 1KB
soft(第一个)-软磁盘空间限制，表示用户可以使用的磁盘空间大小，单位为 KB。可以有 7 天（默认）的超越，过后自动转为硬限制，不限制设置为 0
hard(第一个)-硬配额限制，不能超越，表示用户可以使用的最大磁盘空间，单位为 KB，不限制设置为 0
inodes-当前文件个数
soft(第二个)-软磁盘空间限制，可以有 7 天（默认）的超越，表示用户可以创建的文件个数，包括目录，可以有默认七天的超越
hard(第二个)-硬配额限制，不能超越，表示用户可以创建的文件个数，不能超过。

以上的限制只是对用户设定的硬限制在起作用。如果需要使软限制也起作用的话，还需要对用户的软限制设定宽限期--缺省的，软限制的宽限期是无穷大--这可以使用 edquota 命令的-t 选项来实现

edquota -t
改变宽限期

Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds #可以使用天、小时、分、秒为单位来设定宽限期。

Filesystem	Block grace period	Inode grace period
/dev/fd0	7days	7days

这个只对以后的用户生效,对之前已经过期的用户无效

edquota -p mj `awk -F: '$3 > 499 {print $1}' /etc/passwd`

151 ☆setquota

设置配额

语法:
setquota [-u|-g] 用户名/组名 软块数 硬块数 软文件数 硬文件数 装载点|设备名

setquota -u / 2000 2500 100 110 panda

152 ☆quota

如果不带任何参数运行 quota 的话，查看的是你自己的配额使用情况。

quota -u 用户名 #如果当前磁盘上没有该用户的文件,则不会显示要用-uv 看
查看用户磁盘使用情况
[root@panda nfs]# quota -u root
Disk quotas for user root (uid 0):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/fd0	19*	100	10		3	200	20	

quota -g 组名
看某个组的磁盘使用情况。

如果该用户没有配置磁盘限额的话，输出显示如下：
Disk quotas for user panda (uid 503): none

quota -uv panda
看用户的信息

153 ☆repquota

查看报告,启用限额磁盘上的所有用户报告

```
[root@panda nfs]# repquota -u /mnt/nfs/
*** Report for user quotas on device /dev/fd0
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      19       0       0              3     0     0
```

154 ☆quota 实验

1)修改/etc/fstab 中的某文件系统行，在挂载选项中添加 usrquota,grpquota
/dev/sda5 /mnt/sda5 ext3 defaults,usrquota,grpquota 0 0
简单说明：usrquota 表示支持用户级配额，grpquota 表示支持组级的配额，可以根据需要选择部分或全部。

2)重新挂载刚才修改的文件系统 ，使其支持配额，实际上是修改了 /etc/mtab
#mount -o remount /dev/sda5

3)运行 quotacheck 在支持配额文件系统根下生成配额管理文件(挂载点/aquota.user,aquota.group 两个文件)
#quotacheck -cavug /dev/sda5

4）用 edquota 完成用户或组配额设置/修改
#edquota user1 [-f /dev/sda5] ---修改的是 aquota.user
#edquota -g group1 [-f /dev/sda5] ---修改的是 aquota.group
简单说明：
第一行是设置用户 user1 的配额限制
第二行是设置组 group1 的配额限制
[-f /dev/sda5]表可选，不选表示在所有启用磁盘配额的文件系统上使用相同的配额设置，一般不选即可。

5)打开磁盘配额监控进程
#quotaon -avug [/dev/sda1]

6)显示磁盘配额使用状态
#repquota -a 或 repquota /dev/sda1
#repquota -g -a 或 repquota -a /dev/sda2 (组的配额)

7）暂时关闭某个文件系统的配额
#quotaoff -avug 停止所有所有文件系统的用户和组的配额

8)取消某个文件系统的配额限制
#quotaoff -vug 文件系统
#删除/etc/fstab 中设置配额的部分

9）修改软配额的最大超越时间
#edquota -t [-g] 修改用户/配额软配额超越的最大天数，也就是用户超过 soft 的限制后，系统允许在设定的时间范围内继续超越。默认是 7 天

9）补充说明
/，/boot/,/proc,/mnt/cdrom 等不要使用配额，其实也没用。 而且磁盘配额不适合 FAT 和 FAT32。
以后当新设置了某个用户的配额，可以使用如下命令，马上生效。
#quotacheck -auvgm --是不尝试重新挂载文件系统
使用 edquota -t 可以修改配额软限制的超限时间

删除配额很简单
先 quotaoff
改 fstab 文件
再删除配额数据库

155 ☆cdrecorder

cdrecorder -scanbus

cdrecorder -v speed=4 dev=0,0,0 cd.iso

156 ☆dvdrecorder

-scanbus

157 ☆mkisofs

-r 支持基于 unix 文件系统
-J mircosoft 下可用
-T 长文件名
-o 输出

mkisofs -J -r -T -o cd.iso /home
mount -t iso9660 -o loop cd.iso /mnt/cdrom
isoinfo -i cd.iso -l
显示其中文件

158 ☆mknod

语法:
mknod [OPTION]... NAME TYPE [MAJOR MINOR]

说明:
建立特殊的文件

Mandatory arguments to long options are mandatory for short options too.

-Z, --context=CONTEXT
set security context (quoted string)

-m, --mode=MODE
设置权限

当 TYPE 为 b,c,或者 u 的时候 MAJOR 和 MINOR 必须指定,
当 TYPE 为 p 的时候 MAJOR 和 MINOR 可以忽略,
如果 MAJOR 或 MINOR 以 0x 或 0X 开始,是 16 进制.
0 开始的是 8 进制
其他是 10 进制

TYPE 可以为:

b
建立一个块设备(可缓冲)文件

c,u
建立一个字符设备(不可缓冲)文件

p
建立一个 FIFO

159 ☆mknod 实验

/dev 中有
crw------- 1 root root   10, 63 2006-08-18 device-mapper

total 232
crw-------   1 root root 10,10 Sep 15 2003 adbmouse

每行的第一个字母，例如:c 表示字符设备，如果是 b，则表示块设备。
接下来第一个 10 是主设备号，第二个 63 是次设备号。

在文件系统上创建一个设备节点的命令是 mknod，例如:
mknod /dev/scull0 c 254 0
将创建一个字符设备(c)，主设备号 254，次设备号 0。

//////////////////////////
// 动态分配主设备号 //
//////////////////////////

一部分主设备号已经分配给了大部分常见设备，可以在内核源代码树的 Documentation/devices.txt 文件或者/proc/devices 文件找到:

/proc/devices 文件内容如下:
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  6 lp
  7 vcs
 10 misc
 13 input
 14 sound
 29 fb
 36 netlink
 89 i2c
116 alsa
128 ptm
136 pts
162 raw
180 usb

Block devices:
  1 ramdisk
  2 fd
  8 sd
  9 md

22 ide1
65 sd
66 sd
67 sd
68 sd
69 sd
70 sd
71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
253 device-mapper
254 mdp

1.列出/dev/rhd02，查出他的主、副设备号
2.mknod /dev/hd02 b 主号 副号
如
/dev # l rhd02
crw------- 1 sysinfo sysinfo 1, 23 Jan 24 08:35 rhd02
mknod hd02 b 1 23

字符设备和块设备的设备文件用 mknod 命令创建，用主设备号和次设备号标识，同一个设备驱动程序控制的所有设备具有相同的主设备号，并用不同的次设备号加以区别

160 ☆主设备号和次设备号

如果你执行 ls -l /dev 命令，在设备文件条目中的最新修改日期前你会看到二个数（用逗号分隔），这个位置通常显示文件长度。这二个数就是相应设备的主设备号和次设备号。
主设备号识别设备对应的驱动程序。
次设备号只由相应的设备驱动程序使用；内核的其他部分不使用它，仅将它传递给驱动程序。所以一个驱动程序管理若干个设备并不为奇（如下面的例子所示），次序号提供了一种区分它们的方法。

crw-rw-rw- 1 root root 1, 3 2006-08-18 null
crw-rw-rw- 1 root root 1, 5 2006-08-18 zero
/dev/null 和/dev/zero 都由驱动程序 1 管理
crw------- 1 vcsa tty 7, 1 8 月 17 22:44 vcs1

crw------- 1 vcsa tty 7, 129 8 月 17 22:44 vcsa1
所有的虚拟控制台和串口终端都由驱动程序 4 管理
crw------- 1 root root 4, 1 8 月 17 22:44 tty1
crw-rw---- 1 root uucp 4, 64 8 月 17 22:44 ttyS0
crw-rw---- 1 root uucp 4, 65 8 月 17 22:44 ttyS1
vcs1 和 vcsa1 由驱动程序 7 管理

当设备打开（open）时，内核利用主设备号分派执行相应的驱动程序。

161☆devfs

设备文件系统:devfs
如果使用了这种文件系统，那设备文件将变得简单，但也很原来有很大的不同。
当没有使用 devfs 时，向系统增加一个驱动程序意味着要赋值它一个主设备号。这一赋值过程应该在驱动程序（模块）的初始化过程中完成，它调用如下函数，这个函数定义在<linux/fs.h>：
int register_chrdev(unsigned int major, const char *name, struct file_operations *fops);
返回值提示成功或者失败。返回一个负值，表示出错；返回零或正值，表示成功。参数 major 是所请求的主设备号，name 是你的设备的名字，它将在/proc/devices 中出现，fops 是一个指向函数队列的指针，利用它完成对设备函数的调用，本章稍后将在“文件操作”一节中介绍这些内容。
主设备号是一个用来索引静态字符设备组的整数，“动态分配主设备号”将在本章的稍后部分中介绍怎样选择一个主设备号。2.0 内核支持 128 个设备驱动，而 2.2 和 2.4 内核支持 256 个（保留数值 0 和 255 为将来使用）。而次版本号（8 位字节的数）并没有传递给 register_chrdev 函数，因为次版本号是驱动程序自己使用的。开发团队为了增加内核可能支持的设备数量而带来了很大的压力，在开发树 2.5 版本内核的目标中，设备号至少是 16 位的。
一旦设备驱动程序注册到内核表中，它的操作都与分配的主设备号匹配，何时在字符设备文件上操作都与它的主设备号相关联，内核都会通过 file_operations 结构体查找并调用相应的驱动程序中的函数。为了这个原因，传递给 register_chrdev 的指针应该是指向驱动程序中的全局结构体，而不是一个局部的一个模块初始化函数。
接下来的问题就是如何给程序一个名字以被它们用来请求你的设备驱动程序。这个名字必须插入到/dev 目录中，并与你的驱动程序的主设备号和次设备号相连。
在文件系统上创建一个设备节点的命令是 mknod，而且你必须是超级用户才能操作。除了要创建的节点名字外，该命令还带三个参数。例如，命令：
mknod /dev/scull0 c 254 0
创建一个字符设备（c），主设备号是 254，次设备号是 0。由于历史原因，次设备号应该在 0-255 范围内，有时它们存储在一个字节中。存在很多原因扩展可使用的次设备号的范围，但就现在而言，仍然有 8 位限制。
请注意：如果一旦用 mknod 生成了一个特别的设备文件，它就永远存在了硬盘上，除非你明白的删除了它。你可以通过执行命令 rm /dev/scull0 来删除例子中的设备。
动态分配主设备号
某些主设备号已经静态地分配给了大部分公用设备。在内核源码树的 Documentation/device.txt 文件中可以找到这些设备的列表。由于许多数字已经分配了，为新设备选择一个唯一的号码是很困难的——用户的设备数量要比可用的主设备号多得多。

很幸运（或是更应该感谢某些人的智慧），你可以动态请求分配主设备号。如果你调用 register_chrdev 时的 major 为 0 的话，这个函数就会选择一个空闲号码并做为返回值返回。主设备号总是正的，而如果返回负值的话，就表示是错误码。请注意这二种情况下，操作存在着微小的差别：如果调用者请求动态分配一个号码，则函数返回已经分配好的主设备号。而返回 0 则表明成功分配调用预先指定好的主设备号。
作为个人的设备，我们强烈推荐你使用动态分配机制获取你的主设备号，而不要随便选择一个一个当前不用的设备号做为主设备号。当然，如果你的设备在社会上能大范围的使用，并且已经包含到官方的内核树中，你需要分配一个主设备号作为专用。
动态分配的缺点是，由于分配给你的主设备号不能保证总是一样的，无法事先创建设备节点。这意味着你不能用加载命令加载你的驱动程序，这个高级特性将在 11 章中介绍。对于设备的普通使用，这不是什么问题，这是因为一旦分配了设备号，你就可以从/proc/devices 读到。
为了加载一个使用动态分配主设备号的设备驱动程序，对 insmod 的调用可以用调用 insmod、读/proc/devices 后的一个简单的脚本来代替生成特殊文件。 （To load a driver using a dynamic major number, therefore, the invocation of insmod can be replaced by a simple script that after calling insmod reads /proc/devices in order to create the special file(s).）
/proc/devices 一般如下所示：
Character devices:
1 mem
2 pty
3 ttyp
4 ttyS
6 lp
7 vcs
10 misc
13 input
14 sound
21 sg
180 usb
Block devices:
2 fd
8 sd
11 sr
65 sd
66 sd
加载动态分配主设备号的模块的脚本可以利用象 awk 这类工具来写，该脚本从/proc/devices 中获取信息，并在/dev 中创建文件。
下面这个脚本，scull_load，是 scull 发行中的一部分。使用以模块形式发行的驱动程序的用户可以在系统的 rc.local 文件中调用这个脚本，或是在需要模块时手工调用。

```
#!/bin/sh
module="scull"
device="scull"
mode="664"
```

```
# invoke insmod with all arguments we were passed
# and use a pathname, as newer modutils don't look in . by default
/sbin/insmod -f ./$module.o $* || exit 1
# remove stale nodes
rm -f /dev/${device}[0-3]
major=‘awk "\\$2==\"$module\" {print \\$1}" /proc/devices‘
mknod /dev/${device}0 c $major 0
mknod /dev/${device}1 c $major 1
mknod /dev/${device}2 c $major 2
mknod /dev/${device}3 c $major 3
# give appropriate group/permissions, and change the group.
# Not all distributions have staff; some have "wheel" instead.
group="staff"
grep 'staff:' /etc/group > /dev/null || group="wheel"
chgrp $group /dev/${device}[0-3]
chmod $mode /dev/${device}[0-3]
```

这个脚本同样可以适用于其他驱动程序，只要重新定义变量和调整 mknod 那几行就可以了。上面那个脚本创建 4 个设备，4 是 scull 源码中的默认值。

脚本的最后几行看起来有点古怪：为什么要改变设备的组和模式呢？原因是只有超级用户才能运行这段脚本。默认允许位只允许 root 对其有写访问权，而其他只有读权限。正常情况下，设备节点需要不同的策略，因此某些访问权限需要进行某些修改。虽然这个脚本允许一组用户访问，但你要作一些修改。稍后，在第 5 章的“设备文件的访问控制”一节中的 sculluid 源码，将展示设备驱动程序如何实现自己的设备访问授权。scull_unload 脚本用来整理/dev 目录，并删掉这个模块。作为交替使用加载和卸载的一对脚本，你可以一个初始化脚本，用来放在你发布的目录下。作为 scull 的源码一部分，我们提供了相当完整的和可配置的初始化脚本的例子：scull.init;它接受传统的参数如“start” 、“stop” 、 “restart”，来执行 scull_load 和 scull_unload 的角色。.

如果重复地创建和删除/dev 节点似乎有点过分的话，这里一个有用的解决方法。如果你只是单单加载和卸载一个简单的驱动，你可以在第一次利用脚本生成特殊文件后使用 rmmod 和 insmod 命令来完成，你也可以计算出你选择的号码同时也不用弄乱其他的（动态）模块，对开发避免繁长的脚本是非常有用的。当然，这方法不适合同一时间运行多个驱动的情况。

就我们的观点中，分配主设备号的最佳方式是默认采用动态分配，同时也留有在加载时，甚至是编译时，指定主设备号的选择权。采用我的建议的代码将与自动端口探测的代码十分类似。scull 的实现使用了一个全局变量，scull_major，来保存所选择的设备号。该变量是由在 scull.h 中定义的常数 SCULL_MAJOR 初始化的，该值在所发行的源码中为 0，即“选择动态分配”。用户可以使用这个默认值，也可以指定某个特定的主设备号，在编译前修改宏定义即可，也可以在 ins_mod 命令行中指定 scull_major 的值。最后，通过使用 scull_load 脚本，用户可以使用 scull_load 的命令行中将参数传递给 insmod。

这里是我在 scull.c 源码中的获取主设备号的代码：

```
result = register_chrdev(scull_major, "scull", &scull_fops);
if (result < 0) {
printk(KERN_WARNING "scull: can't get major %d\n",scull_major);
return result;
```

}
if (scull_major == 0) scull_major = result; /* dynamic */
从系统中删除设备驱动程序
当从系统中卸载一个模块时，必需释放主设备号。在清除（cleanup）模块的函数中调用如下函数完成该操作：
int unregister_chrdev(unsigned int major, const char *name);
参数是要释放的主设备号和相应的设备名。内核对这个名字和设备号对应的名字进行比较：如果不同，返回-ENINVAL。如果主设备号超出了所允许的范围或是并未分配给这个设备，内核一样返回-EINVAL。
在清除（cleanup）函数中如注销资源失败会有非常不好的后果。当下次试图读取 /proc/devices 时将产生一个一个错误，是由于其中一个 name 字串仍然指向模块内存，而那片内存已经不存在了。这种失效称为 oops，因为当内核在访问无效地址时将打印这样的消息。
当你卸载驱动程序而又没有注销主设备号时，这种情况将产生很难弥补的过错，即便为此专门写一个“补救”模块也无济于事，因为 unregister_chrdev 中的 trcmp 使用不同与原始模块对应的指针（名字）。当你注销主设备号失败时，你必须同时重新加载原模块和为注销这个主设备号而创建的模块。如果你没有修改过代码，那这个有缺点的模块将幸运地获得同个地址，而名字将存放在这个地址中。当然，作为安全的选择，是重起系统。
除了卸载模块，你还经常需要在卸载驱动程序时删除设备节点。我们用加载模块时使用的一对脚本中的另一个来完成这项工作。对于我们的样例设备，脚本 scull_unload 完成这个工作，作为选择，你也可以调用 scull.init 中的 stop。
如果动态节点没有从/dev 中删除，就会有可能造成不可预期的错误：开发者计算机上的一个没有删除的/dev/framegrabber 就有可能在一个月后引用一个火警设备，如果这二个设备都使用动态获得主设备号。当打开这个 /dev/framegrabber 设备时，产生“没有这个文件或目录”的错误总要比打开一个新设备所产生的后果要好得多。
dev_t 和 kdev_t
到目前为止，我们已经讨论了主设备号。现在是讨论次设备号和驱动程序如何使用次设备号来区分设备的时候了。
每次内核调用一个设备驱动程序时，它都告诉驱动程序它正在操作哪个设备。主设备号和次设备号合在一起构成一个数据类型并用来标别某个设备。组合的设备号（主设备号和次设备号合在一起）保存在 “inode”结构的 i_rdev 域中，inode 将在稍后介绍。一些驱动程序接收一个指向 struct inode 的指针做为第一个参数。这个指针通常也称为 inode（通常驱动开发这也这样称呼），函数可以通过查看 inode->i_rdev 分解出设备号。
历史上，Unix 通过申明 dev_t 变量（设备类型 device type）来保存设备号。dev_t 通常是<sys/types.h>中定义的一个 16 位整数。而现在有时需要超过 256 个次设备号，但是由于有许多应用（包括 C 库在内）都“了解”dev_t 的内部结构，所以改变 dev_t 是很困难的，因为如果改变 dev_t 的内部结构就会造成这些应用无法运行。因此，虽然许多系统（groundwork）已经放置了更大的设备号，但是他们目前仍然只限制使用 16 位整数。
然而，在 Linux 内核内部却使用了一个新类型，kdev_t。对于每一个内核函数来说，这个新类型被设计为一个黑箱。用户程序完成不能知道 kdev_t，而系统也不知道 kdev_t 里边究竟是什么东西。如果 kdev_t 一直是隐藏的，它可以在内核的不同版本间任意变化，而不必修改每个人的设备驱动程序。

有关 kdev_t 的信息被限制在<linux/kdev_t.h>中，其中大部分是注释。如果你对代码背后的原理感兴趣的话，这个文件是前部分有教育性的指导说明。因为<linux/fs.h>已经包含了这个头文件，没有必要显式地包含这个文件。
如下这些宏和函数是你可以对 kdev_t 执行的操作：
MAJOR(kdev_t dev);
从 kdev_t 结构中提取出主设备号。
MINOR(kdev_t dev);
提取出次设备号。
MKDEV(int ma, int mi);
通过主设备号和次设备号生成 kdev_t。
kdev_t_to_nr(kdev_t dev);
将 kdev_t 转换为一个整数（dev_t）。
to_kdev_t(int dev);
将一个整数转换为 kdev_t。注意，核心态中没有定义 dev_t，因此使用了 int。
只要你通过这些函数来操作设备号，那它哪怕只是作为内部数据结构变换，也将继续运行。

162 ☆COM 口符号

com1 口是 ttyS1
以此类推

163 ☆kudzu

硬件检测
可以用在热交换的时候

164 ☆lspci

lspci 列出所有 PCI 设备
lspci - list all PCI devices ，主要是有来列出机器中的 PCI 设备
lspci 是读取 hwdata 数据库，hwdata 由软件包 hwdata 提供;

hwdata 大约有如下文件
[root@panda nfs]# rpm -ql hwdata
/etc/hotplug/blacklist
/etc/pcmcia
/etc/pcmcia/config
/usr/X11R6/lib/X11/Cards
/usr/share/doc/hwdata-0.146.1.EL
/usr/share/doc/hwdata-0.146.1.EL/COPYING
/usr/share/doc/hwdata-0.146.1.EL/LICENSE
/usr/share/hwdata
/usr/share/hwdata/CardMonitorCombos

/usr/share/hwdata/Cards
/usr/share/hwdata/MonitorsDB
/usr/share/hwdata/pci.ids
/usr/share/hwdata/pcitable
/usr/share/hwdata/upgradelist
/usr/share/hwdata/usb.ids

[root@ltest ~]# lspci -
Usage: lspci [<switches>]

-v Be verbose
-n Show numeric ID's
-b Bus-centric view (PCI addresses and IRQ's instead of those seen by the CPU)
-x Show hex-dump of the standard portion of config space
-xxx Show hex-dump of the whole config space (dangerous; root only)
-xxxx Show hex-dump of the 4096-byte extended config space (root only)
-s [[[[<domain>]:]<bus>]:][<slot>][.[<func>]] Show only devices in selected slots
-d [<vendor>]:[<device>] Show only selected devices
-t Show bus tree
-m Produce machine-readable output
-i <file> Use specified ID database instead of /usr/share/hwdata/pci.ids
-M Enable `bus mapping' mode (dangerous; root only)
-P <dir> Use specified directory instead of /proc/bus/pci
-H <mode> Use direct hardware access (<mode> = 1 or 2)
-F <file> Read configuration data from given file
-G Enable PCI access debugging

lspci 有两个参数是我们常用的，-b 和-v
lspci -b,所有设备清单,也会把 usb 接口列出来
lspci -v 来查看更为详细的内容
我们可以通过 lspci -v 来查看硬件的 irq 中断什么的;比如网卡不好用，是不是 irq 存在冲突了

165☆dmesg

dmesg 是一个显示内核缓冲区系统控制信息的工具;比如系统在启动时的信息会写到/var/log/
注:dmesg 工具并不是专门用来查看硬件芯片组标识的工具，但通过这个工具能让我们知道机器中的硬件的一些参数;因为系统在启动的时候，会写一些硬件相关的日志到 /var/log/message* 或 /var/log/boot* 文件中;

如果我们用这个工具来查看一些硬件的信息;这个工具信息量太大，的确需要耐心;
[root@panda ~]# dmesg
[root@panda ~]# dmesg -c 注:清理掉缓冲区，下次开机的时候还会自动生成;

166 ☆hwbrowser

hwbrowser 是您当前硬件配置的图形化浏览器 ，这个工具是图形的。可能系统在默认的情况下没有安装。需要您安装才行。
[root@panda ~]# rpm -ivh hwbrowser*.rpm

[root@panda ~]# hwbrowser

167 ☆lshal 和 hal-device-manager

通过 lshal 和 hal-device-manager 也能知道硬件相关信息，不过这个工具对新手操作起来是有点麻烦
hwbrowser 是 lshal 的图形化界;可能系统在默认的情况下没有安装

[root@panda ~]# yum install hal-device-manager
或
[root@panda ~]# apt install hal-device-manager

[root@panda ~]# hal-device-manager

[root@panda ~]# lshal

168 ☆硬件配置文件和工具

显示器和显示卡的配置文件是/etc/X11/xorg.conf ，对于老的 Linux 版本应该是/etc/X11/X86Config 或 X86Config-4 ，xorg 或 X86 的配置文件还包括鼠标、键盘等在 X 桌面环境下的配置和控制等;
声卡的配置文件是/etc/modprobe.conf 或 /etc/modules.conf ;
文件系统的配置文件是 /etc/fstab ;
声卡的配置工具，一般情况下是 alsaconf

当然不同的发行版也有相应的硬件配置或管理工具，比如 Fedora Core 4.0 有如下的硬件配置或管理工具;
[root@panda ~]# system-config-mouse
[root@panda ~]# system-config-network-tui
[root@panda ~]# gnome-system-monitor
[root@panda ~]# system-config-network
[root@panda ~]# system-config-printer-gui
[root@panda ~]# system-config-soundcard
[root@panda ~]# setup
当然这些图形化的工具并不是万能的，他们的功能还是有极大的局限性;所以在硬件配置上，大多还是通过文本模式来操作的;

硬件驱动如果是以内核模块支持的，驱动目录位于: /lib/modules/内核版本/kernel/ 目录 或 /lib/modules/内核版本/kernel/drivers 目录中;

[root@panda ~]# uname -r
2.6.11-1.1369_FC4
[root@panda ~]# ls /lib/modules/2.6.11-1.1369_FC4/kernel
arch crypto drivers fs lib net sound

注:只有驱动在内核中以模块的方法支持的，或者我们自己安装的驱动，驱动才位于 /lib/modules/ 相应的目录;如果是直接置入内核的，不会出现在/lib/modules 驱动相关的目录;

169 ☆各重要命令与文件的 RPM 包出处表

（RHEL4 版）
1. initscripts-7.93.11.EL-1 : /etc/inittab, /etc/rc*, /etc/sysconfig/*, /sbin/ifup,/sbin/ifdown,/sbin/redhat-support-check,/sbin/initlog, etc.
2.util-linux-2.12a-16.EL4.6 : /bin/mount, /bin/login, /bin/umount,/bin/kill, /etc/pam.d/login,/sbin/fdisk, etc.
3.filesystem-2.3.0-1 : /initrd, /proc, /root, etc. All filesystem related.
4.SysVinit-2.85-34 : /sbin/halt, /sbin/init, /sbin/poweroff, /sbin/reboot, /sbin/shutdown, /sbin/telinit, etc.
5.kernel-utils-2.4-13.1.48, kernel-devel-2.6.9-5.EL, kernel-2.6.9-5.EL, etc. All kernel related packages contain RHEL4 kernel and tools.
6.Setup-2.5.37-1.1 : /etc/aliases, /etc/bashrc, /etc/environment, /etc/exports, /etc/filesystems, /etc/group, /etc/passwd, /etc/hosts.allow, /etc/hosts.deny, /etc/profile, /etc/services, /etc/securetty, /etc/shells, /etc/printcap, /etc/inputrc, etc.
7.bash-3.0-19.2 : /bin/bash, /bin/sh, /etc/skel/*, etc.
8.rpm-4.3.3-7_nonptl and all rpm related packages contain rpm related binaries.
9.Grub-0.95-3.1 : /boot/grub/*, /sbin/grub*, /usr/share/grub/*, etc.
10. /etc/fstab 和 /boot/grub/grub.conf 或 menu.lst 不属于任何一个包，是系统安装后产生的，所以必须手动修复！！

170 ☆system-config-packages

system-config-packages --tree=/mnt/inst
指定源来安装包

171 ☆RPM

RPM 是 Red Hat Package Manager 的缩写，本意是 Red Hat 软件包管理，顾名思义是 Red Hat 贡献出来的软件包管理;在 Fedora 、Redhat、Mandriva、SuSE、YellowDog 等主流发行版本，以及在这些版本基础上二次开发出来的发行版采用;

RPM 包里面都包含什么？里面包含可执行的二进制程序，这个程序和 Windows 的软件包中的.exe 文件类似是可执行的;RPM 包中还包括程序运行时所需要的文件，这也和 Windows 的软件包类似，Windows 的程序的运行，除了.exe 文件以外，也有其它的文件;

RPM 包管理的用途;
1、可以安装、删除、升级和管理软件;当然也支持在线安装和升级软件;
2、通过 RPM 包管理能知道软件包包含哪些文件，也能知道系统中的某个文件属于哪个软件包;
3、可以在查询系统中的软件包是否安装以及其版本;
4、作为开发者可以把自己的程序打包为 RPM 包发布;
5、软件包签名 GPG 和 MD5 的导入、验证和签名发布
6、依赖性的检查，查看是否有软件包由于不兼容而扰乱了系统;

RPM 的使用权限;
RPM 软件的安装、删除、更新只有 root 权限才能使用;对于查询功能任何用户都可以操作;如果普通用户拥有安装目录的权限，也可以进行安装;

Redhat 系统管理软件包工具 redhat-config-packages
Fedora 系统管理软件包工具 system-config-packages

rpm 软件包的文件名中，不仅包含了软件名称，版本信息，还包括了适用的硬件架构的信息。
比如 mypackage-1.1-2TL.i386.rpm，其中 mypackage 是在系统中登记的软件包的名字
1.1 是软件的版本号，2 是发行号，TL 表示用于 TL 操作系统，还可能是 RH 等。
i386 表示用于 intel x86 平台，还可能是 sparc 等。
i386 指这个软件包适用于 intel 80386 以上的 x86 架构的计算机(AI32)
i686 指这个软件包适用于 intel 80686 以上(奔腾 pro 以上)的 x86 架构的计算机(IA32)
noarch 指这个软件包于硬件架构无关，可以通用。
i686 软件包的程序通常针对 CPU 进行了优化，所以，向后兼容比较用以，i386 的包在 x86 机器上都可以用。向前一般不兼容。不过现在的计算机，奔腾 pro 以下的 CPU 已经很少用，通常配置的机器都可以使用 i686 软件包

172 ☆RPM 安装

命令格式：
rpm -i (or --install) options file1.rpm ... fileN.rpm
这个是用来安装一个新的 rpm 包;

参数：
file1.rpm ... fileN.rpm 将要安装的 RPM 包的文件名

详细选项：

-h (or --hash) 安装时输出 hash 记号 (``#")
--test 只对安装进行测试，并不实际安装。
--percent 以百分比的形式输出安装的进度。
--excludedocs 不安装软件包中的文档文件
--includedocs 安装文档
--replacepkgs 强制重新安装已经安装的软件包
--replacefiles 替换属于其它软件包的文件
--force 忽略软件包及文件的冲突
--noscripts 不运行预安装和后安装脚本
--prefix <path> 将软件包安装到由 <path> 指定的路径下
--ignorearch 不校验软件包的结构
--ignoreos 不检查软件包运行的操作系统
--nodeps 不检查依赖性关系
--ftpproxy <host> 用 <host> 作为 FTP 代理
--ftpport <port> 指定 FTP 的端口号为 <port>

通用选项
-v 显示附加信息
-vv 显示调试信息
--root <path> 让 RPM 将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile <rcfile> 设置 rpmrc 文件为 <rcfile>
--dbpath <path> 设置 RPM 资料库存所在的路径为 <path>

安装过程中可能出现下面的警告或者提示:
... conflict with ...
可能是要安装的包里有一些文件可能会覆盖现有的文件，缺省时这样的情况下是无法正确安装的
可以用
rpm --force -i 强制安装即可

... is needed by ...
... is not installed ...
此包需要的一些软件你没有安装可以用
rpm --nodeps -i 来忽略此信息

rpm -i --force --nodeps
可以忽略所有依赖关系和文件问题，什么包都能安装上，但这种强制安装的软件包不能保证完全发挥功能

参数--replacepkgs 是以已安装的软件再安装一次;有时没有太大的必要;
rpm -ivh --replacepkgs lynx-2.8.5-23.i386.rpm

测试安装参数 --test ，用来检查依赖关系;并不是真正的安装;
rpm -ivh --test gaim-1.3.0-1.fc4.i386.rpm

为软件包指定安装目录:要加 --relocate 参数;下面的举例是把 gaim-1.3.0-1.fc4.i386.rpm 指定安装在 /opt/gaim 目录中;
rpm -ivh --relocate /=/opt/gaim gaim-1.3.0-1.fc4.i386.rpm
这样重定义目录后,我们安装在指定目录中的程序如何调用呢？一般执行程序，都放在安装目录的 bin 或者 sbin 目录中;看下面的例子;如果有错误输出，就做相应的链接，用 ln -s ;

--aid 应该是自动安装有依赖关系的包
--aid 是需要建立本地的软件包数据库，导入数字签名之后才能用

173 ☆RPM 删除

命令格式：
rpm -e (or --erase) options pkg1 ... pkgN

参数
pkg1 ... pkgN ： 要删除的软件包

详细选项
--test 只执行删除的测试
--noscripts 不运行预安装和后安装脚本程序
--nodeps 不检查依赖性

通用选项
-vv 显示调试信息
--root <path> 让 RPM 将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile <rcfile> 设置 rpmrc 文件为 <rcfile>
--dbpath <path> 设置 RPM 资料库存所在的路径为 <path>

删除一个 rpm 包
[root@panda ~]#rpm -e 软件包名

包名可以包含版本号等信息，
rpm -e lynx
注意:不可以有后缀.rpm
用 rpm -e **.rpm 是无法删除 rpm 包

有时会出现一些错误或者警告:
... is needed by ...

这说明这个软件被其他软件需要，不能随便卸载可以用
rpm -e --nodeps 强制卸载,忽略依赖的检查来删除

但尽可能不要这么做，最好用软件包管理器 systerm-config-packages 来删除或者添加软件

174 ☆RPM 升级

命令格式
rpm -U (or --upgrade) options file1.rpm ... fileN.rpm
这是用来升级一个 rpm 包

参数
file1.rpm ... fileN.rpm 软件包的名字

详细选项
-h (or --hash) 安装时输出 hash 记号 (``#")
--oldpackage 允许"升级"到一个老版本
--test 只进行升级测试
--excludedocs 不安装软件包中的文档文件
--includedocs 安装文档
--replacepkgs 强制重新安装已经安装的软件包
--replacefiles 替换属于其它软件包的文件
--force 忽略软件包及文件的冲突
--percent 以百分比的形式输出安装的进度。
--noscripts 不运行预安装和后安装脚本
--prefix <path> 将软件包安装到由 <path> 指定的路径下
--ignorearch 不校验软件包的结构
--ignoreos 不检查软件包运行的操作系统
--nodeps 不检查依赖性关系
--ftpproxy <host> 用 <host> 作为 FTP 代理
--ftpport <port> 指定 FTP 的端口号为 <port>

通用选项
-v 显示附加信息
-vv 显示调试信息
--root <path> 让 RPM 将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile <rcfile> 设置 rpmrc 文件为 <rcfile>
--dbpath <path> 设置 RPM 资料库存所在的路径为 <path>

rpm -Uvh file.rpm --nodeps --force

由新版本降级为旧版本，要加 --oldpackage 参数;
rpm -Uvh --oldpackage gaim-1.3.0-1.fc4.i386.rpm

rpm {-U|--upgrade} [install-options] PACKAGE_FILE ...
新包安装后旧版本都删除

rpm {-F|--freshen} [install-options] PACKAGE_FILE ...
可以指定 ftp 或者 http 做源,也是更新包功能同上

175 ☆RPM 查询

命令格式：
rpm {-q|--query} [select-options] [query-options]

参数：
pkg1 ... pkgN ： 查询已安装的软件包

详细选项
-p <file>(or ``-") 查询软件包的文件
-f <file> 查询<file>属于哪个软件包
-a 查询所有安装的软件包
--whatprovides <x> 查询提供了 <x>功能的软件包
-g <group> 查询属于<group> 组的软件包
--whatrequires <x> 查询所有需要 <x> 功能的软件包

信息选项
<null> 显示软件包的全部标识
-i 显示软件包的概要信息
-l 显示软件包中的文件列表
-c 显示配置文件列表
-d 显示文档文件列表
-s 显示软件包中文件列表并显示每个文件的状态
--scripts 显示安装、卸载、校验脚本
--queryformat (or --qf) 以用户指定的方式显示查询信息
--dump 显示每个文件的所有已校验信息
--provides 显示软件包提供的功能
--requires (or -R) 显示软件包所需的功能

通用选项
-v 显示附加信息
-vv 显示调试信息

--root <path> 让 RPM 将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile <rcfile> 设置 rpmrc 文件为 <rcfile>
--dbpath <path> 设置 RPM 资料库存所在的路径为 <path>

查看系统中所有已经安装的包，要加 -a 参数
rpm -qa

查询与该包管理系统相关的软件包
rpm -qa |grep 软件名

检查安装了多少个包
rpm -qa | wc -l

查询系统已安装的软件
语法:rpm -q 软件名

查询已安装软件包都安装到何处
语法:rpm -ql 软件名 或 rpm rpmquery -ql 软件名

rpm -ql lynx
rpmquery -ql lynx

查询一个已安装软件包的信息
语法格式: rpm -qi 软件名

rpm -qi lynx

查看一下已安装软件的配置文件;
语法格式:rpm -qc 软件名

rpm -qc lynx

查看一个已经安装软件的文档安装位置
语法格式: rpm -qd 软件名

rpm -qd lynx

查询一个已经安装的文件属于哪个软件包
语法:rpm -qf 文件名
注:文件名所在的绝对路径要指出

检查哪个包提供了 vimtutor 文件
rpm -qf `which vimtutor`

列出 coreutils 提供的文件和相关信息
rpm -qil coreutils

某个程序是哪个软件包安装的，或者哪个软件包包含这个程序
rpm -qf `which 程序名` 返回软件包的全名
rpm -qif `which 程序名` 返回软件包的有关信息
rpm -qlf `which 程序名` 返回软件包的文件列表
注意，这里不是引号，而是`，就是键盘左上角的那个键。
也可以使用 rpm -qilf，同时输出软件包信息和文件列表

查看一下已安装软件所依赖的软件包及文件
语法格式: rpm -qR 软件名

rpm -qR rpm-python

对于未安装的软件包的查看

查看的前提是您有一个.rpm 的文件，也就是说对既有软件 file.rpm 的查看等;

查看一个软件包的用途、版本等信息
语法: rpm -qpi file.rpm

rpm -qpi lynx-2.8.5-23.i386.rpm

查看一件软件包所包含的文件
语法: rpm -qpl file.rpm

rpm -qpl lynx-2.8.5-23.i386.rpm

查看软件包的文档所在的位置;
语法: rpm -qpd file.rpm

rpm -qpd lynx-2.8.5-23.i386.rpm

查看一个软件包的配置文件;
语法: rpm -qpc file.rpm

rpm -qpc lynx-2.8.5-23.i386.rpm

查看一个软件包的依赖关系
语法: rpm -qpR file.rpm

rpm -qpR yumex_0.42-3.0.fc4_noarch.rpm

总结:
获得软件包相关的信息用 rpm -q，q 表示查询 query，后面可以跟其他选项，比如
i 表示 info，获得软件包的信息;
l 表示 list，获得文件列表;
a 表示 all，在所有包中执行查询;
f 表示 file，根据文件进行相关的查询;
p 表示 package，根据软件包进行查询
需要的查询条件可以使用 grep 产生，或者从"` `"中的命令行产生

176 ☆RPM 导入签名

rpm --import 签名文件

rpm --import RPM-GPG-KEY

177 ☆RPM 校验已安装的软件包

命令格式:
rpm -V (or --verify, or -y) options

参数
pkg1 ... pkgN 将要校验的软件包名

软件包选项
-p <file> Verify against package file <file>
-f <file> 校验<file>所属的软件包
-a Verify 校验所有的软件包
-g <group> 校验所有属于组 <group> 的软件包

详细选项
--noscripts 不运行校验脚本
--nodeps 不校验依赖性
--nofiles 不校验文件属性

通用选项
-v 显示附加信息
-vv 显示调试信息

--root <path> 让 RPM 将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile <rcfile> 设置 rpmrc 文件为 <rcfile>
--dbpath <path> 设置 RPM 资料库存所在的路径为 <path>

--verify 都是同 rpm 数据库检查的

rpm --verify -a
是检查所有包的完整性

rpm --verify -p /mnt/cdrom/RedHat/RPMS/inn-2.3.5-5.i386.rpm
检查某一个包

rpm --verify --file /bin/ls
检查某个文件

rpm --verify --file /bin/vi
  ......G. /bin/vi

参数含义
Failure Code Meaning
5 MD5 checksum
S File size
L Symbolic link
T File modification time
D Device
U User
G Group
M Mode

rpm -Vf /bin/mount
验证文件是否被改动
无输出是正常的

178 ☆RPM 校验软件包的完整性

语法：
rpm -K (or --checksig) options file1.rpm ... fileN.rpm

参数：
file1.rpm ... fileN.rpm 软件包的文件名

Checksig--详细选项

--nopgp 不校验 PGP 签名

通用选项
-v 显示附加信息
-vv 显示调试信息
--rcfile <rcfile> 设置 rpmrc 文件为 <rcfile>

rpm -K /opt/gaim gaim-1.3.0-1.fc4.i386.rpm
验证完整性

导入签名后就可以检查某个包的完整性了
rpm --checksig /mnt/cdrom/RedHat/RPMS/pkg-1.2.3-4.noarch.rpms

179 ☆RPM 其它 RPM 选项

--rebuilddb 重建 RPM 资料库
--initdb 创建一个新的 RPM 资料库
--quiet 尽可能的减少输出
--help 显示帮助文件
--version 显示 RPM 的当前版本

通过 rpm 命令查询一个 rpm 包是否安装了，也是要通过 rpm 数据库来完成的;所以我们要经常用下面的两个命令来初始化 rpm 数据库;
[root@panda ~]# rpm --initdb
[root@panda ~]# rpm --rebuilddb 注:这个要花好长时间;
注:这两个参数是极为有用，有时 rpm 系统出了问题，不能安装和查询，大多是这里出了问题;

180 ☆RPM 管理包管理器支持网络安装和查询

rpm 参数 rpm 包文件的 http 或者 ftp 的地址

rpm -qpi http://mirrors.kernel.org/fedora/core/4/i386/os/ Fedora/RPMS/gaim-1.3.0-1.fc4.i386.rpm
rpm -ivh http://mirrors.kernel.org/fedora/core/4/i386/os/ Fedora/RPMS/gaim-1.3.0-1.fc4.i386.rpm
rpm -ivh ftp://xxxxxxxx/gaim-1.3.0-1.fc4.i386.rpm

181 ☆从 RPM 软件包抽取文件

命令格式: rpm2cpio file.rpm |cpio -div

[root@localhost RPMS]# rpm2cpio gaim-1.3.0-1.fc4.i386.rpm |cpio -div

抽取出来的文件就在当前操作目录中的 usr,var 和 etc 中;

其实这样抽到文件不如指定安装目录来安装软件来的方便;也一样可以抽出文件;

为软件包指定安装目录:要加 --relocate 参数;下面的举例是把 gaim-1.3.0-1.fc4.i386.rpm 指定安装在 /opt/gaim 目录中;

```
[root@localhost RPMS]# rpm -ivh --relocate /=/opt/gaim gaim-1.3.0-1.fc4.i386.rpm
Preparing... ########################################### [100%]
   1:gaim ########################################### [100%]
[root@localhost RPMS]# ls /opt/
gaim
```

这样也能一目了然;gaim 的所有文件都是安装在 /opt/gaim 中，我们只是把 gaim 目录备份一下，然后卸掉 gaim;这样其实也算提取文件的一点用法;

使用工具 rpm2cpio 和 cpio

```
rpm2cpio xxx.rpm | cpio -vi
rpm2cpio xxx.rpm | cpio -idmv
rpm2cpio xxx.rpm | cpio --extract --make-directories
```

参数 i 和 extract 相同，表示提取文件。
v 表示指示执行进程
d 和 make-directory 相同，表示根据包中文件原来的路径建立目录
m 表示保持文件的更新时间。

```
rpm2cpio file.rpm |cpio -it
```

查看列表

```
rpm2cpio file.rpm |cpio -imd ./etc/samba/smb.conf
```

提取文件

182 ☆RPM 的配置文件

RPM 包管理，的配置文件是 rpmrc

```
[root@localhost RPMS]# locate rpmrc
/usr/lib/rpm/rpmrc
/usr/lib/rpm/redhat/rpmrc
```

```
rpm --showrc
```

也可以查看

183 ☆编译源代码包的条件

首先我们在 Linux 系统中至少得把开发工具安装上，比如 gcc ;perl;python;glibc;gtk;make ;automake 等开发工具或基础包;还要安装一些相应的开发包，一般是文件名包括 dev 的，比如 kernel-devel;还有一些开发库，比如以 lib 开头的;如果您在编译软件时，有时提示缺少什么东西之类的，大多少的是这些开发工具和开发库等;从光盘中找出安装就是了;有时光盘没有提供，请用 google 搜索相应的软件包，有时可能也会用到源码包编译安装所依赖的包;

有时本来系统中已经安装了所依赖的包，但系统提示找不到应该怎么办？这时需要我们设置一下 PKG_CONFIG_PATH 的环境变量就行了;
#export PKG_CONFIG_PATH=/usr/lib/pkgconfig
或
#export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
然后我们再来运行编译的./configure ;make ;make install ，尝试着来吧;

以 java 开发的工具开发的程序，要用到 jre 或者 jdk ;jdk 已经包括 jre 了，所以如果我们只是要求有一个 java 程序运行的环境，只需要安装 jre 就行了;安装好 jre，配置一下 java 的环境变量就可以用了。如果是图形界面的程序，点点鼠标就 OK 了;

用 perl 开发的程序，是需要 perl 环境的，所以必须得把 perl 的包安装上，python 也同理;

184 ☆安装源代码包

源代码一般以 file.tar.gz file.tar.bz2 或 file.src.rpm 打包;file.tar.gz 和 file.tar.bz2 格式
这种软件包里面都是源程序，没有编译过，需要编译后才能安装，安装方法为:
[root@panda ~]# tar jxvf file.tar.bz2(或 bz)
[root@panda ~]# tar zxvf file.tar.gz

大多以 tar.gz 和 tar.bz2 打包软件，大多是通过 ./configure ;make ;make install 来安装的;有的软件是直接 make;make install ;
通过./configure --help 来查看配置软件的功能;
大多软件是提供./configure 配置软件的功能的;少数的也没有，如果没有的就不用./configure ;直接 make;make install 就行了;
./configure 比较重要的一个参数是 --prefix ，用--prefix 参数，我们可以指定软件安装目录;当我们不需要这个软件时，直接删除软件的目录就行了;

用 CD 命令进入解压缩后的目录
输入编译文件命令: ./configure(有的压缩包已经编译过，这一步可以省去)
然后是命令: make
再是安装文件命令: make install
安装完毕

源代码包卸载
用 CD 命令进入编译后的软件目录，即安装时的目录
执行反安装命令:make uninstall

指定源码包安装目录
比如我们可以指定 fcitx 安装到 /opt/fcitx 目录中;
[root@localhost fcitx]#./configure --prefix=/opt/fcitx
如果我们不需要 fcitx 时，可以直接删除 /opt/fcitx 目录;
一般的软件的默认安装目录在/usr/local 或者/opt 里，可以到那里去找找

如果您想要让 fcitx 只要执行 fcitx ，就能调用，请配置环境变量，或者在/usr/bin 中做一个 fcitx 的链接;
[root@panda ~]# ln -s /opt/fcitx/bin/fcitx /usr/bin/fcitx

185 ☆src.rpm 软件包

有些软件包是以.src.rpm 结尾的，这类软件包是包含了源代码的 rpm 包，在安装时需要进行编译。
这类软件包有两种安装方法，
方法一:
1.执行 rpm -i your-package.src.rpm
2. cd /usr/src/redhat/SPECS
3. rpmbuild -bp your-package.specs 一个和你的软件包同名的 specs 文件
4. cd /usr/src/redhat/BUILD/your-package/ 一个和你的软件包同名的目录
5. ./configure 这一步和编译普通的源码软件一样，可以加上参数
6. make
7. make install

方法二:
1.执行 rpm -i you-package.src.rpm
2. cd /usr/src/redhat/SPECS
前两步和方法一相同
3. rpmbuild -bb your-package.specs 一个和你的软件包同名的 specs 文件
这时，在/usr/src/redhat/RPM/i386/ (根据具体包的不同，也可能是 i686,noarch 等等)
在这个目录下，有一个新的 rpm 包，这个是编译好的二进制文件。
执行 rpm -i new-package.rpm 即可安装完成。

rpmbuild -ba redhat-config-packages.spec
一步一步完成

rpmbuild -bb redhat-config-packages.spec
直接建立二进制文件

#rpmbuild --rebuild mplayer-0.90pre8.20021004-1.src.rpm
#cd /usr/src/redhat(or other)/RPMS/i386/
#rpm -ivh xxxxx.rpm
上面的方法是对 Redhat 8.0 以及 Mandrake9.0 来说的，如果是低于这个版本。应该用
#rpm --rebuild *.src.rpm

186 ☆以 bin 结尾的安装包

用 CD 命令进入源代码压缩包所在的目录
给文件加上可执行属性:chmod +x ******.bin
执行命令:./******.bin
以 bin 结尾的安装包的卸载:把安装时中选择的安装目录删除就 OK

187 ☆基于 perl 和 python 的程序的安装

一般情况下，用
#perl file.pl 安装

perl vmware-install.pl

基于 python 开发，用
python file.py
来安装

188 ☆脚本型安装程序

要用
#sh 文件名

sh NFORCE-Linux-x86-1.0-0306-pkg1.run

当然也能通过 chmod 755 NFORCE-Linux-x86-1.0-0306-pkg1.run ，然后 ./NFORCE-Linux-x86-1.0-0306-pkg1.run 来安装;

189 ☆源码目录

Directory Purpose
/usr/src/redhat/SOURCES 包括程序的源代码
/usr/src/redhat/SPECS 包括 spec 文件(控制 RPM 编译过程)
/usr/src/redhat/BUILD 在这里解压和编译
/usr/src/redhat/RPMS 包括输出的二进制 RPM
/usr/src/redhat/SRPMS 包括 SRPM,由编译过程产生

190 ☆文件系统权限

Linux 系统中的每个文件和目录都有访问许可权限，用它来确定谁可以通过何种方式对文件和目录进行访问和操作。

用-l 参数的 ls 命令显示文件的详细信息
ls -l

显示几个栏位为:
文件类型与权限 硬链接数 文件属主 文件属组 文件大小 建立或最近修改的时间 名字

对于符号链接文件，显示的文件名之后有"->"和引用文件路径名。

对于设备文件，其"文件大小"字段显示主、次设备号，而不是文件大小。目录中的总块数显示在长格式列表的开头，其中包含间接块。

用 ls -l 命令显示的信息中，开头是由 10 个字符构成的字符串，其中第一个字符表示文件类型，它可以是下述类型之一:
- 普通文件
d 目录
l 符号链接
b 块设备文件
c 字符设备文件
s 套接口文件
p 管道

后面的 9 个字符表示文件的访问权限，分为 3 组，每组 3 位。第一组表示文件属主的权限，第二组表示同组用户的权限，第三组表示其他用户的权限。每一组的三个字符分别表示对文件的读、写和执行权限。

第 2~10 个字符当中的每 3 个为一组，左边三个字符表示所有者权限，中间 3 个字符表示与所有者同一组的用户的权限，右边 3 个字符是其他用户的权限。这三个一组共 9 个字符，代表的意义如下：
- 没有设置权限。
r(Read，读取)：对文件而言，具有读取文件内容的权限;对目录来说，具有浏览目录的权限。
w(Write,写入)：对文件而言，具有新增、修改文件内容的权限;对目录来说，具有删除、移动目录内文件的权限。
x(eXecute，执行)：对文件而言，具有执行文件的权限;对目录了来说该用户具有进入目录的权限。
s 或 S（SUID,Set UID）：可执行的文件搭配这个权限，便能得到特权，任意存取该文件的所有者能使用的全部系统资源。请注意具备 SUID 权限的文件，黑客经常利用这种权限，以 SUID 配上 root 帐号拥有者，无声无息地在系统中开扇后门，供日后进出使用。suid 位对目录不起什么作用，

可以将一条只有 root 用户才可以使用的命令提供给普通执行，例如：/sbin/ifconfig 这个命令，注意，当普通用户要执行设置了 suid 的命令
时必须要加完全路径，/sbin/ifconfig，这是因为和 root 用户的 PATH 的环境变不同导致的。
s 或 S（SGID，Set GID）：设置在文件上面，其效果与 SUID 相同，只不过将文件所有者换成用户组，该文件就可以任意存取整个用户组所能使用的系统资源。sgid 位对目录的作用是将此目录下所有新建的文件所属的组都自动设置成该目录所属的组，无论创建者的组是什么
t 或 T（Sticky）：设置标志位(留在内存，不被换出)。如果该文件是目录，在该目录中的文件只能被超级用户、目录拥有者或文件属主删除。如果它是可执行文件，在该文件执行后，指向其正文段的指针仍留在内存。这样再次执行它时，系统就能更快地装入该文件。/tmp 和 /var/tmp 目录供所有用户暂时存取文件，亦即每位用户皆拥有完整的权限进入该目录，去浏览、删除和移动文件。粘滞位的作用是该目录下创建的文件,只有所有者可以更名，删除，其他用户即使拥有完全访问权限也无权删除其他用户的文件。
粘贴位现在也很少用了，不过对于象/tmp 目录这样的，是整个系统临时文件存放在，还是有点意义。一个目录既使它的所有权限都开放 rwxrwxrwx，如果是设置了粘帖位，除非目录的属主和 root 用户有权限删除它，除此之外其它用户不能删除这个目录。用途一般是把一个文件夹的的权限都打开，然后来共享文件，象/tmp 目录一样。

因为 SUID、SGID、Sticky 占用 x 的位置来表示，所以在表示上会有大小写之分。加入同时开启执行权限和 SUID、SGID、Sticky，则权限表示字符是小写的：
如果关闭执行权限，则表示字符会变成大写：

有三种不同类型的用户可对文件或目录进行访问:文件所有者，同组用户、其他用户。所有者一般是文件的创建者。所有者可以允许同组用户有权访问文件，还可以将文件的访问权限赋予系统中的其他用户。在这种情况下，系统中每一位用户都能访问该用户拥有的文件或目录。

每一文件或目录的访问权限都有三组，每组用三位表示，分别为文件属主的读、写和执行权限;与属主同组的用户的读、写和执行权限;系统中其他用户的读、写和执行权限。当用 ls -l 命令显示文件或目录的详细信息时，最左边的一列为文件的访问权限。

[root@localhost ~]# ls -lih
总计 104K
2408949 -rwxr-xr-x 1 root root 7 04-21 12:47 lsfile.sh
2408830 drwxr-xr-x 2 root root 4.0K 04-21 12:46 mkuml-2004.07.17
2408260 drwxr-xr-x 2 root root 4.0K 04-21 22:15 mydir
2408258 lrwxrwxrwx 1 root root 7 04-21 22:16 sun001.txt -> sun.txt
2408263 -rw-r--r-- 2 root root 11 04-20 14:17 sun002.txt
2408263 -rw-r--r-- 2 root root 11 04-20 14:17 sun.txt

解释：
第一字段：inode
第二字段：文件种类和权限;
第三字段：硬链接个数;

第四字段：文件属主;
第五字段：所归属的组;
第六字段：文件或目录的大小;
第七字段和第八字段：最后访问或修改时间;
第九字段：文件名或目录名

权限位
每个文件或目录都有一组 9 个权限位,每三位被分为一组,他们分别是属主权限位(占三个位置)、用户组权限位（占三个位置）、其它用户权限位（占三个位置）。比如 rwxr-xr-x ，我们数一下就知道是不是 9 个位置了，正是这 9 个权限位来控制文件属主、用户组以及其它用户的权限。

Linux 文件或目录的权限位是由 9 个权限位来控制，每三位为一组，它们分别是文件属主(Ower)的读、写、执行，用户组(Group)的读、写、执行以及(Other)其它用户的读、写、执行;

文件属主：读 r、写 w、执行 x
用户组：读 r、写 w、执行 x
其它用户：读 r、写 w、执行 x

如果权限位不可读、不可写、不可执行，是用-来表示。

对于普通文件的读、写、执行权限可以这样理解：
可读：意味着我们可以查看阅读;
可写：意味着，可以修改或删除（不过删除或修改的权限受父目录上的权限控制）;
可执行：意味着如果文件就可以运行，比如二进制文件（比如命令），或脚本（要用脚本语言解释器来解释运行）。

查看文件的属性用 ls -l 文件 ;查看目录的属性是 ls -d 目录。

191 ☆权限实验

setuid 和 setgid 的解说 ;
setuid 和 setgid 位是让普通用户可以以 root 用户的角色运行只有 root 帐号才能运行的程序或命令。比如我们用普通用户运行 passwd 命令来更改自己的口令，实际上最终更改的是/etc/passwd 文件。我们知道/etc/passwd 文件是用户管理的配置文件，只有 root 权限的用户才能更改。
[root@localhost ~]# ls -l /etc/passwd
-rw-r--r-- 1 root root 2379 04-21 13:18 /etc/passwd

作为普通用户如果修改自己的口令通过修改/etc/passwd 肯定是不可完成的任务，但是不是可以通过一个命令来修改呢。答案是肯定的，作为普通用户可以通过 passwd 来修改自己的口令。这归功于 passwd 命令的权限。我们来看一下;
[root@localhost ~]# ls -l /usr/bin/passwd
-r-s--x--x 1 root root 21944 02-12 16:15 /usr/bin/passwd

因为/usr/bin/passwd 文件已经设置了 setuid 权限位（也就是 r-s--x--x 中的 s），所以普通用户能临时变成 root，间接的修改/etc/passwd，以达到修改自己口令的权限。

setuid 和 setgid 的实例应用;
我们想让一个普通用户 panda 拥有 root 用户拥有超级 rm 删除权限，我们除了用 su 或 sudo 临时切换到 root 身份操作以外，还能怎么做呢？？？

[root@localhost ~]#cd /home 注：进入/home 目录
[root@localhost home]# touch pandatest.txt 注：创建一个测试文件;
[root@localhost home]# ls -l pandatest.txt 注：查看文件属性;
-rw-r--r-- 1 root root 0 04-24 18:03 pandatest.txt 注：文件的属性;
[root@localhost home]# su panda 注：切换到普通用户 panda
[panda@localhost home]$ rm -rf pandatest.txt 注：以普通用户身份来删除 pandatest.txt 文件;
rm: 无法删除 “pandatest.txt”：权限不够

那我们怎么才能让 panda 这个普通用户也拥有 root 超级的 rm 删除功力呢？

[root@localhost ~]# ls -l /bin/rm
-rwxr-xr-x 1 root root 93876 02-11 14:43 /bin/rm
[root@localhost ~]# chmod 4755 /bin/rm 注：设置 rm 的权限为 4755 ， 就把 setuid 位设置好了。
[root@localhost ~]# ls -l /bin/rm
-rwsr-xr-x 1 root root 43980 02-11 14:43 /bin/rm
[root@localhost ~]# cd /home/
[root@localhost home]# su panda 注：切换到 panda 用户身份;
[root@localhost home]$ ls -l pandatest.txt 注：查看文件属性;
-rw-r--r-- 1 root root 0 04-24 18:03 pandatest.txt 注：文件的属性;
[panda@localhost home]$ rm -rf pandatest.txt 注：删除 pandatest.txt 文件;

我们只是设置了 rm 的 setuid 位，让普通用户在 rm 指令上有超级 root 的删除超级权力。

文件属主和属组的特殊情况 ;
[root@localhost ~]# ls -lh sungood.txt
-rw-r--r-- 1 501 502 85 04-25 13:45 sungood.txt

上面的例子是不是有点怪？因为他的属主和属组都是一个数值;这是为什么呢？出现这种情况的原因是系统中不存在与之对应的用户，所以只能以数字形式显示了。有时我们删除了用户，但没有删除其家目录，这种情况下，它的家目录的属主和属组也会变成数字;

[root@localhost ~]# userdel linuxsir
[root@localhost ~]# ls -ld /home/linuxsir

drwx------ 16 501 502 4096 03-27 02:28 /home/linuxsir

影响文件的读写执行的因素;
一个文件能不能被读取，要受到它的属主、属组及其它用户权限的影响，还要受到其父目录权限的影响。我们来举个例子;
[root@localhost ~]# cd /home 注：进入/home 目录;
[root@localhost home]# mkdir redhatdir 注：创建一个目录 redhatdir
[root@localhost home]# touch redhatdir/test.txt 注：创建一个文件 test.txt
[root@localhost home]# chmod 700 redhatdir/ 注：修改 redhatdir 的权限 ，为属主可读可写可执行，属组和其它用户无权限;
[root@localhost home]# ls -ld redhatdir/ 注：查看 redhatdir 的属性;
drwx------ 2 root root 4096 04-25 13:01 redhatdir/
[root@localhost home]# ls -lr redhatdir/ 注：查看 test.txt 文件的属性;
总计 0
-rw-r--r-- 1 root root 0 04-25 13:02 test.txt
[root@localhost home]# su panda 注：我们切换到普通用户 panda
[panda@localhost home]$ cd redhatdir/ 注：进入 redhatdir 目录，以 panda 用户身份。
bash: cd: redhatdir/: 权限不够
[panda@localhost home]$ more redhatdir/test.txt
redhatdir/test.txt: 权限不够

解释：我们通过这个例子来看，为什么 test.txt 在其它用户权位上拥有可读权限 r--，但我们用普通用户还不能查看它的内容呢？这是因为他的父目录没有其它用户的何读权限。我们是不是 redhatdir 目录的其它用户可读权限打开，就能让普通用户 panda 能读取 test.txt 的内容了呢？？
[root@localhost home]# chmod 704 redhatdir/
[root@localhost home]# ls -ld redhatdir/
drwx---r-- 2 root root 4096 04-25 13:02 redhatdir
[root@localhost home]# su panda
[panda@localhost home]$ cd redhatdir/
bash: cd: redhatdir/: 权限不够

看来如果不设置属组的权限，只打开属主的权限及其它用户在 redhatdir 目录的读权限的情况下，其它用户是不能访问的;我们应该把 test.txt 父目录的 redhatdir 的属主的读、写、执行要打开，还要把父目录的属组的读和执行权限打开，其它用户的读和执行权限打开，也就是要拥有 rwxr-xr-x 权限，这样文件的其它用户才能访问。
[root@localhost home]# chmod 755 redhatdir/
[root@localhost home]# more redhatdir/test.txt

192 ☆linux 常见目录

/ Linux 文件系统的入口，也是处于最高一级的目录;

/bin 基础系统所需要的那些命令位于此目录，也是最小系统所需要的命令;比如 ls、cp、mkdir 等命令;功能和/usr/bin 类似，这个目录中的文件都是可执行的，普通用户都可以使用的命令。做为基础系统所需要的最基础的命令就是放在这里。

/boot Linux 的内核及引导系统程序所需要的文件，比如 vmlinuz initrd.img 文件都位于这个目录中。在一般情况下，GRUB 或 LILO 系统引导管理器也位于这个目录;

/dev 设备文件存储目录，比如声卡、磁盘... ...

/etc 系统配置文件的所在地，一些服务器的配置文件也在这里;比如用户帐号及密码配置文件;

/home 普通用户家目录默认存放目录;

/lib 库文件存放目录

/lost+found 在 ext2 或 ext3 文件系统中，当系统意外崩溃或机器意外关机，而产生一些文件碎片放在这里。当系统启动的过程中 fsck 工具会检查这里，并修复已经损坏的文件系统。 有时系统发生问题，有很多的文件被移到这个目录中，可能会用手工的方式来修复，或移到文件到原来的位置上。

/media 即插即用型存储设备的挂载点自动在这个目录下创建，比如 USB 盘系统自动挂载后，会在这个目录下产生一个目录 ;CDROM/DVD 自动挂载后，也会在这个目录中创建一个目录，类似 cdrom 的目录。这个只有在最新的发行套件上才有，比如 Fedora Core 4.0 5.0 等。可以参看/etc/fstab 的定义;

/misc

/mnt 这个目录一般是用于存放挂载储存设备的挂载目录的，比如有 cdrom 等目录。可以参看 /etc/fstab 的定义。有时我们可以把让系统开机自动挂载文件系统，把挂载点放在这里也是可以的。主要看/etc/fstab 中怎么定义了;比如光驱可以挂载到/mnt/cdrom 。

/opt 表示的是可选择的意思，有些软件包也会被安装在这里，也就是自定义软件包，比如在 Fedora Core 5.0 中，OpenOffice 就是安装在这里。有些我们自己编译的软件包，就可以安装在这个目录中;通过源码包安装的软件，可以通过 ./configure --prefix=/opt/目录 。

/proc 操作系统运行时，进程（正在运行中的程序）信息及内核信息（比如 cpu、硬盘分区、内存信息等）存放在这里。/proc 目录伪装的文件系统 proc 的挂载目录，proc 并不是真正的文件系统，它的定义可以参见 /etc/fstab 。

/root Linux 超级权限用户 root 的家目录;

/sbin 大多是涉及系统管理的命令的存放，是超级权限用户 root 的可执行命令存放地，普通用户无权限执行这个目录下的命令，这个目录和/usr/sbin;/usr/X11R6/sbin 或/usr/local/sbin 目录是相似的;我们记住就行了，凡是目录 sbin 中包含的都是 root 权限才能执行的。

/tmp 临时文件目录，有时用户运行程序的时候，会产生临时文件。/tmp 就用来存放临时文件的。/var/tmp 目录和这个目录相似。

/usr 这个是系统存放程序的目录，比如命令、帮助文件等。这个目录下有很多的文件和目录。当我们安装一个 Linux 发行版官方提供的软件包时，大多安装在这里。如果有涉及服务器配置文件的，会把配置文件安装在/etc 目录中。/usr 目录下包括涉及字体目录/usr/share/fonts ，帮助目录/usr/share/man 或/usr/share/doc，普通用户可执行文件目录/usr/bin 或/usr/local/bin 或/usr/X11R6/bin ，超级权限用户 root 的可执行命令存放目录，比如 /usr/sbin 或/usr/X11R6/sbin 或/usr/local/sbin 等;还有程序的头文件存放目录/usr/include。

/var 这个目录的内容是经常变动的，看名字就知道，我们可以理解为 vary 的缩写，/var 下有/var/log 这是用来存放系统日志的目录。/var/www 目录是定义 Apache 服务器站点存放目录;/var/lib 用来存放一些库文件，比如 MySQL 的，以及 MySQL 数据库的的存放地;

/etc/init.d 这个目录是用来存放系统或服务器以 System V 模式启动的脚本，这在以 System V 模式启动或初始化的系统中常见。比如 Fedora/RedHat;

/etc/xinit.d 如果服务器是通过 xinetd 模式运行的，它的脚本要放在这个目录下。有些系统没有这个目录， 比如 Slackware，有些老的版本也没有。在 Rehat/Fedora 中比较新的版本中存在。

/etc/rc.d 这是 Slackware 发行版有的一个目录，是 BSD 方式启动脚本的存放地;比如定义网卡，服务器开启脚本等。

/etc/X11 是 X-Windows 相关的配置文件存放地;

/usr/bin 这个目录是可执行程序的目录，普通用户就有权限执行; 当我们从系统自带的软件包安装一个程序时，他的可执行文件大多会放在这个目录。比如安装 gaim 软件包时。相似的目录是/usr/local/bin; 有时/usr/bin 中的文件是/usr/local/bin 的链接文件;

/usr/sbin 这个目录也是可执行程序的目录，但大多存放涉及系统管理的命令。只有 root 权限才能执行;相似目录是/sbin 或/usr/local/sbin 或/usr/X11R6/sbin 等;

/usr/local 这个目录一般是用来存放用户自编译安装软件的存放目录;一般是通过源码包安装的软件，如果没有特别指定安装目录的话，一般是安装在这个目录中。这个目录下面有子目录。自己看看吧。

/usr/lib 和/lib 目录相似，是库文件的存储目录;

/usr/share 系统共用的东西存放地，比如 /usr/share/fonts 是字体目录，是用户都共用的吧。

/usr/share/doc 和/usr/share/man 帮助文件，也是共用的吧;

/usr/src 是内核源码存放的目录，比如下面有内核源码目录，比如 linux 、linux-2.xxx.xx 目录等。有的系统也会把源码软件包安装在这里。比如 Fedora/Redhat，当我们安装 file.src.rpm 的时候，这些软件包会安装在/usr/src/redhat 相应的目录中。请参考： 《file.src.rpm 使用方法的简单介绍》 。另外 Fedhat 4.0 5.0，他的内核源码包的目录位于/usr/src/kernels 目录下的某个目录中（只有安装后才会生成相应目录）;

/var/adm 比如软件包安装信息、日志、管理信息等，在 Slackware 操作系统中是有这个目录的。在 Fedora 中好象没有;自己看看吧。
/var/log 系统日志存放，分析日志要看这个目录的东西;
/var/spool 打印机、邮件、代理服务器等假脱机目录;

193 ☆Linux 的特殊块设备

RAM Disk device:
/dev/ram1 ~ /dev/ram8
device number (1,1)-(1,8) 缺省 ramdisk 最大为 4M, 其实际容量根据占用情况调整.若要使用大于 4M 的 ramdisk,可以给 kernel 加参数 ramdisk_size 来调整 eg: mke2fs /dev/ram1 在块设备/dev/ram1 上建 ext2 fs mount /dev/ram1 /mnt mke2fs /dev/ram2 2048 设定最大为 2048K mount /dev/ram2 /mnt

Loopback Device device:
/dev/loop0 ~ /dev/loop7
device number (7,0)-(7,7)
所谓 loopback device 指的就是拿文件来模拟块设备, 首先你的 kernel 要支持, 可以在 compile kernel 时加入 loop 支持，或加入 loop 模块 insmod loop eg: dd if=/dev/zero of=/tmp/rootfs bs=1k count=2048 建一个 2M 的全 0 文件 mke2fs /tmp/rootfs 在 rootfs 文件上建 ext2 fs mount /tmp/rootfs /mnt -o loop 将 rootfs mount 在/mnt 上 这里介绍的是简单的-o loop 用法，另一种用法是 losetup(man losetup for more details) eg: losetup /dev/loop1 /tmp/rootfs 将 loop1 设备与 rootfs 相连接, 此时 losetup /dev/loop1 可见信息 mount /dev/loop1 /mnt ... losetup -d /dev/loop1 去除 loop1 设备与文件的确联结

MD (Multiple Devices) device:
/dev/md0 ~ /dev/md3
device number (9,0)-(9,3)
MD driver 可以将几个硬盘分区合起来成为一个逻辑上的块设备, 现在支持 linear(线性相连)和 RAID-0(使数据尽可能均匀分布在各分区上).内核中一般都已有 MD support

194 ☆/proc

proc - process information pseudo-filesystem
proc 文件系统是一个伪文件系统，它只存在内存当中，而不占用外存空间。它以文件系统的方式为访问系统内核数据的操作提供接口。用户和应用程序可以通过 proc 得到系统的信息，并可以改变内核的某些参数。由于系统的信息，如进程，是动态改变的，所以用户或应用程序读取 proc 文件时，proc 文件系统是动态从系统内核读出所需信息并提交的。它的目录结构如下：
目录名称 目录内容
apm 高级电源管理信息
cmdline 内核命令行
Cpuinfo 关于 Cpu 信息
Devices 可以用到的设备（块设备/字符设备）
Dma Used DMS channels
Filesystems 支持的文件系统
Interrupts 中断的使用
Ioports I/O 端口的使用
Kcore 内核核心印象
Kmsg 内核消息
Ksyms 内核符号表
Loadavg 负载均衡
Locks 内核锁
Meminfo 内存信息
Misc Miscellaneous
Modules 加载模块列表
Mounts 加载的文件系统
Partitions 系统识别的分区表
Rtc Real time clock
Slabinfo Slab pool info
Stat 全面统计状态表 s
Swaps 对换空间的利用情况
Version 内核版本
Uptime 系统正常运行时间

并不是所有这些目录在你的系统中都有，这取决于你的内核配置和装载的模块。另外，在/proc 下还有三个很重要的目录：net，scsi 和 sys。
Sys 目录是可写的，可以通过它来访问或修改内核的参数，而 net 和 scsi 则依赖于内核配置。例如，如果系统不支持 scsi，则 scsi 目录不存在。

除了以上介绍的这些，还有的是一些以数字命名的目录，它们是进程目录。系统中当前运行的每一个进程都有对应的一个目录在/proc 下，以进程的 PID 号为目录名，它们是读取进程信息的接口。而 self 目录则是读取进程本身的信息接口，是一个 link。Proc 文件系统的名字就是由之而起。进程目录的结构如下：

目录名称 目录内容
Cmdline 命令行参数
Environ 环境变量值
Fd 一个包含所有文件描述符的目录
Mem 进程的内存被利用情况
Stat 进程状态
Status Process status in human readable form
Cwd 当前工作目录的链接
Exe Link to the executable of this process
Maps 内存印象
Statm 进程内存状态信息
Root 链接此进程的 root 目录

修改内核参数
在/proc 文件系统中有一个有趣的目录：/proc/sys。它不仅提供了内核信息，而且可以通过它修改内核参数，来优化你的系统。但是你必须很小心，因为可能会造成系统崩溃。最好是先找一台无关紧要的机子，调试成功后再应用到你的系统上。
要改变内核的参数，只要用 vi 编辑或 echo 参数重定向到文件中即可。下面有一个例子：
cat /proc/sys/fs/file-max
4096
echo 8192 > /proc/sys/fs/file-max
cat /proc/sys/fs/file-max
8192
如果你优化了参数，则可以把它们写成脚本文件，使它在系统启动时自动完成修改

proc 文件系统做为内核 kernel 数据结构的接口，把 kernel 的一些信息(比如硬件信息，包括 CPU 、网卡、显示卡、内存、文件系统、SCSI 设备 ....)写到 proc 文件系统中，proc 被 mount 到 /proc 目录;/proc 目录中有大数据大多文件是只读的，但一些数据是根据内核的变化而变化的;/proc 目录中的数据是经常变动的，对于系统中的每个进程都有一个 PID;都可以在/proc 中找到;我们也可以通过 ps -aux |more 来查看进程;

我们可以通过 cat 命令来读取/proc 目录下的文件，比如 cpu 的信息;
[root@panda ~]# cat /proc/cpuinfo

195 ☆/proc/partitions

[panda@localhost ~]$ cat /proc/partitions
major minor #blocks name
3 0 78150744 hda
3 1 6144831 hda1
3 2 16386300 hda2

3 5 8193118 hda5
3 6 10241406 hda6
3 7 787153 hda7
3 8 11719386 hda8
3 9 8787523 hda9
8 0 58605120 sda
8 1 58604528 sda1
查看目前机器中的所有磁盘及分区情况

196 ☆/etc 目录文件介绍

3. 网络配置文件
3.1 /etc/hosts
#/etc/hosts
#文件格式: IPaddress hostname aliases
#文件功能: 提供主机名到 IP 地址的对应关系，建议将自己经常使用的主机
加入此文件中，也可将没有 DNS 记录的机器加入到此文件中，
这样会方便网络应用
127.0.0.1 localhost localhost.localdomain
202.118.66.81 helius.dlut.edu.cn helius

3.2 /etc/resolv.conf
文件功能：DNS 客户机配置文件，设置 DNS 服务器的 IP 地址及 DNS 域名
相关文件：/etc/host.conf
文件格式：
domainname 域名
search 域名
nameserver Primary_DNS_Server_IP_address
nameserver Second_DNS_Server_IP_address
其中 domainname 和 search 可同时存在，也可只有一个;nameserver 可指定多个

示例文件内容：
search dlut.edu.cn
nameserver 202.118.66.6

3.3 /etc/host.conf
功能：指定主机名查找方法，通常指先查找文件/etc/hosts,找不到时再向 DNS 服务器请求。
对于大多数用户不用改动此文件内容。

Linux: /etc/host.conf 文件内容
order hosts, bind
multi on

Soalris: /etc/nsswitch.conf 中的 hosts 项
hosts files, dns
3.4 /etc/HOSTNAME (Linux Redhat 5.x Distribution)
功能：设置主机名，不同 LINUX 之间可能有所差别，请使用
egrep hostname /etc/rc.d/init.d/*
或
egrep hostname /etc/init.d/*
查找相应版本上的主机名设置文件及方法。

Linux Redhat 5.x 对应文件: /etc/sysconfig/network 的 HOSTNAME 项。

3.5 /etc/inetd.conf

Internet 超级服务器, 相关程序: /usr/sbin/inetd
相应服务:
telnet
ftp
pop3
r* rsh rcp rlogin (建议最好关闭 r 服务)
其他服务最好关掉

```
#
# inetd.conf This file describes the services that will be available
# through the INETD TCP/IP super server. To re-configure
# the running INETD process, edit this file, then send the
# INETD process a SIGHUP signal.
#
# Version: @(#)/etc/inetd.conf 3.10 05/27/93
#
# Authors: Original taken from BSD UNIX 4.3/TAHOE.
# Fred N. van Kempen,
#
# Modified for Debian Linux by Ian A. Murdock
#
# Modified for RHS Linux by Marc Ewing
#
#
#
# Echo, discard, daytime, and chargen are used primarily for testing.
#
# To re-read this file after changes, just do a 'killall -HUP inetd'
#
```

```
#echo stream tcp nowait root internal
#echo dgram udp wait root internal
#discard stream tcp nowait root internal
#discard dgram udp wait root internal
#daytime stream tcp nowait root internal
#daytime dgram udp wait root internal
#chargen stream tcp nowait root internal
#chargen dgram udp wait root internal
#
# These are standard services.
#
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
gopher stream tcp nowait root /usr/sbin/tcpd gn

# do not uncomment smtp unless you *really* know what you are doing.
# smtp is handled by the sendmail daemon now, not smtpd. It does NOT
# run from here, it is started at boot time from /etc/rc.d/rc#.d.
#smtp stream tcp nowait root /usr/bin/smtpd smtpd
#nntp stream tcp nowait root /usr/sbin/tcpd in.nntpd
#
# Shell, login, exec and talk are BSD protocols.
#
shell stream tcp nowait root /usr/sbin/tcpd in.rshd
login stream tcp nowait root /usr/sbin/tcpd in.rlogind
#exec stream tcp nowait root /usr/sbin/tcpd in.rexecd
talk dgram udp wait root /usr/sbin/tcpd in.talkd
ntalk dgram udp wait root /usr/sbin/tcpd in.ntalkd
#dtalk stream tcp waut nobody /usr/sbin/tcpd in.dtalkd
#
# Pop and imap mail services et al
#
pop-2 stream tcp nowait root /usr/sbin/tcpd ipop2d
pop-3 stream tcp nowait root /usr/sbin/tcpd ipop3d
imap stream tcp nowait root /usr/sbin/tcpd imapd
#
# The Internet UUCP service.
#
#uucp stream tcp nowait uucp /usr/sbin/tcpd /usr/lib/uucp/uucico -l
#
# Tftp service is provided primarily for booting. Most sites
# run this only on machines acting as "boot servers." Do not uncomment
```

```
# this unless you *need* it.
#
#tftp dgram udp wait root /usr/sbin/tcpd in.tftpd
#bootps dgram udp wait root /usr/sbin/tcpd bootpd
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers." Many sites choose to disable
# some or all of these services to improve security.
#
# cfinger is for GNU finger, which is currently not in use in RHS Linux
#
finger stream tcp nowait root /usr/sbin/tcpd in.fingerd
#cfinger stream tcp nowait root /usr/sbin/tcpd in.cfingerd
#systat stream tcp nowait guest /usr/sbin/tcpd /bin/ps -auwwx
#netstat stream tcp nowait guest /usr/sbin/tcpd /bin/netstat -f inet
#
# Time service is used for clock syncronization.
#
time stream tcp nowait nobody /usr/sbin/tcpd in.timed
time dgram udp wait nobody /usr/sbin/tcpd in.timed
#
# Authentication
#
auth stream tcp nowait nobody /usr/sbin/in.identd in.identd -l -e -o
#
# End of inetd.conf

linuxconf stream tcp wait root /bin/linuxconf linuxconf --http
```

3.6 inetd.conf 相关文件/etc/services(SYSV/BSD/LINUX 相同)
基本不用编辑,linux 自带的已包含大部分服务, Solaris 可能需要增加(POP3),
参考相应的服务器安装说明，在此文件中列出了所有可用的网络服务。

```
#
# services This file describes the various services that are
# available from the TCP/IP subsystem. It should be
# consulted instead of using the numbers in the ARPA
# include files, or, worse, just guessing them.
#
```

```
# Version: @(#)/etc/services 2.00 04/30/93
#
# Author: Fred N. van Kempen,
#
# 文件格式：服务名称 端口号/协议 服务别名
tcpmux 1/tcp # rfc-1078
echo 7/tcp
echo 7/udp
discard 9/tcp sink null
discard 9/udp sink null
systat 11/tcp users
daytime 13/tcp
daytime 13/udp
netstat 15/tcp
qotd 17/tcp quote
chargen 19/tcp ttytst source
chargen 19/udp ttytst source
ftp-data 20/tcp
ftp 21/tcp
telnet 23/tcp
smtp 25/tcp mail
time 37/tcp timserver
time 37/udp timserver
rlp 39/udp resource # resource location
name 42/udp nameserver
whois 43/tcp nicname # usually to sri-nic
domain 53/tcp
domain 53/udp
mtp 57/tcp # deprecated
bootps 67/udp # bootp server
bootpc 68/udp # bootp client
tftp 69/udp
gopher 70/tcp # gopher server
rje 77/tcp
finger 79/tcp
http 80/tcp # www is used by some broken
www 80/tcp # progs, http is more correct
link 87/tcp ttylink
kerberos 88/udp kdc # Kerberos authentication--udp
kerberos 88/tcp kdc # Kerberos authentication--tcp
supdup 95/tcp # BSD supdupd(8)
hostnames 101/tcp hostname # usually to sri-nic
```

iso-tsap 102/tcp
x400 103/tcp # ISO Mail
x400-snd 104/tcp
csnet-ns 105/tcp
pop-2 109/tcp # PostOffice V.2
pop-3 110/tcp # PostOffice V.3
pop 110/tcp # PostOffice V.3
sunrpc 111/tcp
sunrpc 111/tcp portmapper # RPC 4.0 portmapper UDP
sunrpc 111/udp
sunrpc 111/udp portmapper # RPC 4.0 portmapper TCP
auth 113/tcp ident # User Verification
sftp 115/tcp
uucp-path 117/tcp
nntp 119/tcp usenet # Network News Transfer
ntp 123/tcp # Network Time Protocol
ntp 123/udp # Network Time Protocol
netbios-ns 137/tcp nbns
netbios-ns 137/udp nbns
netbios-dgm 138/tcp nbdgm
netbios-dgm 138/udp nbdgm
netbios-ssn 139/tcp nbssn
imap 143/tcp # imap network mail protocol
NeWS 144/tcp news # Window System
snmp 161/udp
snmp-trap 162/udp
exec 512/tcp # BSD rexecd(8)
biff 512/udp comsat
login 513/tcp # BSD rlogind(8)
who 513/udp whod # BSD rwhod(8)
shell 514/tcp cmd # BSD rshd(8)
syslog 514/udp # BSD syslogd(8)
printer 515/tcp spooler # BSD lpd(8)
talk 517/udp # BSD talkd(8)
ntalk 518/udp # SunOS talkd(8)
efs 520/tcp # for LucasFilm
route 520/udp router routed # 521/udp too
timed 525/udp timeserver
tempo 526/tcp newdate
courier 530/tcp rpc # experimental
conference 531/tcp chat
netnews 532/tcp readnews

netwall 533/udp # -for emergency broadcasts
uucp 540/tcp uucpd # BSD uucpd(8) UUCP service
klogin 543/tcp # Kerberos authenticated rlogin
kshell 544/tcp cmd # and remote shell
new-rwho 550/udp new-who # experimental
remotefs 556/tcp rfs_server rfs # Brunhoff remote filesystem
rmonitor 560/udp rmonitord # experimental
monitor 561/udp # experimental
pcserver 600/tcp # ECD Integrated PC board srvr
mount 635/udp # NFS Mount Service
pcnfs 640/udp # PC-NFS DOS Authentication
bwnfs 650/udp # BW-NFS DOS Authentication
kerberos-adm 749/tcp # Kerberos 5 admin/changepw
kerberos-adm 749/udp # Kerberos 5 admin/changepw
kerberos-sec 750/udp # Kerberos authentication--udp
kerberos-sec 750/tcp # Kerberos authentication--tcp
kerberos_master 751/udp # Kerberos authentication
kerberos_master 751/tcp # Kerberos authentication
krb5_prop 754/tcp # Kerberos slave propagation
listen 1025/tcp listener RFS remote_file_sharing
nterm 1026/tcp remote_login network_terminal
kpop 1109/tcp # Pop with Kerberos
ingreslock 1524/tcp
tnet 1600/tcp # transputer net daemon
cfinger 2003/tcp # GNU finger
nfs 2049/udp # NFS File Service
eklogin 2105/tcp # Kerberos encrypted rlogin
krb524 4444/tcp # Kerberos 5 to 4 ticket xlator
irc 6667/tcp # Internet Relay Chat
dos 7000/tcp msdos

End of services.
linuxconf 98/tcp # added by linuxconf RPM

3.7 /etc/hosts.allow /etc/hosts.deny (Linux 下,或使用了 tcpd, 参考 inetd.conf)

/etc/hosts.allow 设置允许使用 inetd 服务的机器，如: All:202.118 即允许所有来自 202.118.x.x 的请求
/etc/hosts.deny 设置不允许使用 inetd 的机器

这两个文件的设定顺序请参考在线文档：
man tcpd

man hosts.allow
man hosts.deny

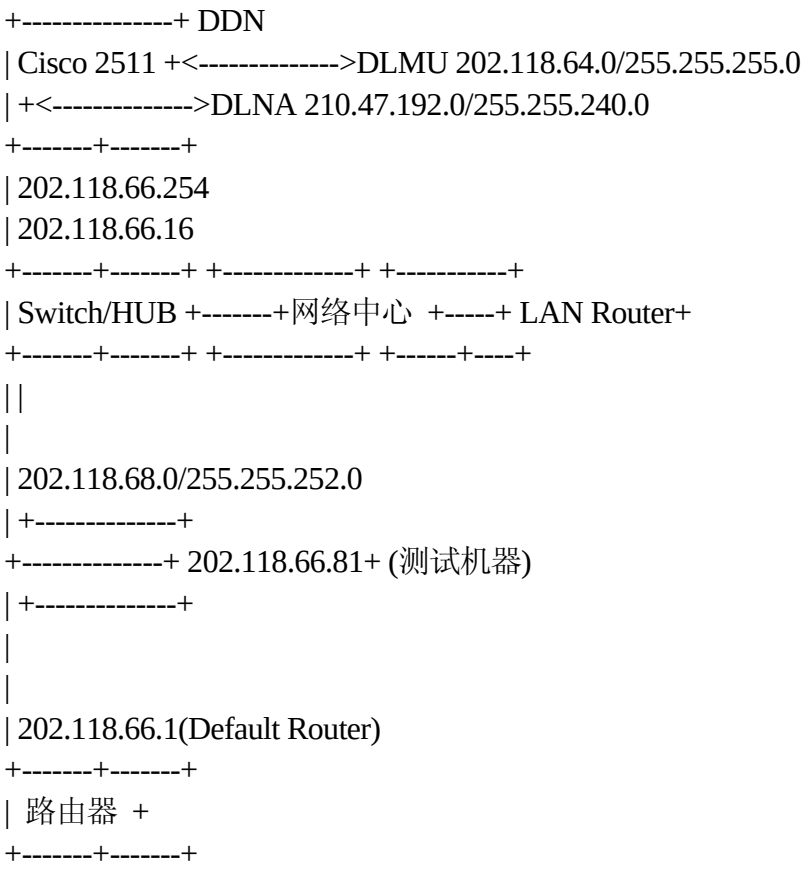
Internet 网络服务访问控制文件,

对于安全性要求较高的服务器建议采用 xinetd 替代 inetd,
xinetd debian 自带,其他的可以用源代码进行编译安装

3.8 /etc/networks /etc/netmasks

列出路由所需要的网络地址,相关命令/usr/sbin/route，当然也可以不使用这两个文件，在维护路由表时可直接使用 IP 地址及网络屏蔽位。

Example:
/etc/networks
dlrin 202.199.128.0
/etc/netmasks
202.199.128.0 255.255.240.0
加入静态路由表项:



|202.112.30.65/255.255.255.252
| DDN
| PPP
|
|202.112.30.66/255.255.255.252
Cernet/Internet

(1) 202.118.66.81(Helius) <-> 202.118.66.18 (peony)

202.118.066.081
255.255.255.0 And

202.118.066.0 网络地址 在同一个 ip 网络段

IP Address <-> MAC(Media Access Address)
202.118.66.18 08:00:20:96:01:6A
202.118.66.81 00:80:C8:4C:6A:D0
202.118.66.1 00:60:5C:F3:FF:75

202.118.66.81 -> 202.118.66.18
以太网的数据包:

08:00:20:96:01:6A + 00:80:C8:4C:6A:D0 + ip 数据

(2) 202.118.66.81 -> 202.112.0.36
不在同一个 ip 段, 通过间接传送(通过路由器).
[hbwork@linden hbwork]$ netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
202.118.66.0 0.0.0.0 255.255.255.0 U 1500 0 0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 U 3584 0 0 lo
0.0.0.0 202.118.66.1 0.0.0.0 UG 1500 0 0 eth0
^^^^^^^
Default Router

(3) 加入静态路由
相关命令：/usr/sbin/route 或 /sbin/route

linux 下需要加入自己网络的路由表项

/sbin/route add -net 202.118.66.0 netmask 255.255.255.0 eth0

/sbin/route add -net 202.199.128.0 netmask 255.255.240.0 gw 202.118.66.254
Or:
/sbin/route add -net dlrin gw 202.118.66.254
Or:
/sbin/route add -net dlrin gw dlrin-gw
/sbin/route add default gw 202.118.66.1

9. /etc/passwd
用户口令文件
10. /etc/shadow (如果有此文件,系统支持 shadow 机制)
$ls -l /etc/shadow
-rwx------ root .... /etc/shadow

11. /etc/fstab
File System Table

#设备名 MountPoint Filesystem Type 加载选项 ... fsck 标志
/dev/hda1 / ext2 defaults 1 1
/dev/hda6 /home ext2 defaults 1 2
/dev/hda3 /usr ext2 defaults 1 2
/dev/hda5 /var ext2 defaults 1 2
/dev/hda2 swap swap defaults 0 0
/dev/fd0 /mnt/floppy auto sync,user,noauto,nosuid,nodev,unhide 0
0
/dev/cdrom /mnt/cdrom auto user,noauto,nosuid,nodev,ro 0 0
none /proc proc defaults 0 0

Solairs 下对应文件: /etc/vfstab

12. /etc/exports

NFS(Network File System) Server 输出文件系统表, 最好不使用 NFS.

nfs 相关进程:
Solaris: mountd , nfsiod
/etc/init.d/nfs.server
Linux: 内核支持 nfs, /proc/filesystem, 也可以通过加载 modules 实现,

13./etc/defaultrouter (Solaris 2.x)
内容为 Default Router 的 ip 地址,
在 linux 下:
Redhat 5.x: /etc/sysconfig/network
GATEWAY=202.118.66.1
GATEWAYDEV=eth0
Debian: /etc/init.d/network
#! /bin/sh
ifconfig lo 127.0.0.1
route add -net 127.0.0.0
IPADDR=202.118.66.88
NETMASK=255.255.255.0
NETWORK=202.118.66.0
BROADCAST=202.118.66.255
GATEWAY=202.118.66.1
ifconfig eth0 ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}
route add -net ${NETWORK}
["${GATEWAY}"] && route add default gw ${GATEWAY} metric 1

14. /etc/bashrc /etc/csh.cshrc /etc/profile
/etc/bashrc BASH(Bourne Again Shell) RunTime Command
Shell Script 用的最多

系统用户默认的环境设置, PATH, umask, TERM Type
/etc/csh.cshrc CSH Runtime COmmand

15. /etc/ftpaccess
FTP 访问控制文件, 文件位置可变 , 通过
#egrep ftp /etc/inetd.conf
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
^^^^^^
ftp 服务器守护进程文件名
#which in.ftpd
/usr/sbin/in.ftpd
#strings /usr/sbin/in.ftpd |egrep ftpaccess
/etc/ftpaccess

相关配置在 ftp 服务器配置中讲述。

16. /etc/ftpusers
不允许 ftp 的用户列表,一般包括 root, uucp, bin 等

17. /etc/ftpconvions /etc/ftpgroups
FTP 服务器配置文件

18. /etc/group 用户组文件

19. /etc/sendmail.cf (Linux) Sendmail(EMAIL 服务器)配置文件
/etc/sendmail.cw 本地主机名
主机名: gingko.dlut.edu.cn
希望接收: user@gingko.dlut.edu.cn
user@mail.dlut.edu.cn
user@dlut.edu.cn
/etdc/aliases 邮件别名文件
/etc/aliases.db 邮件别名二进制数据文件, 用 newaliases 建立
/etc/sendmail.hf sendmail 帮助文件,
$telnet mailserver 25
Trying 202.118.66.8...
Connected to gingko.
Escape character is '^]'.
220 gingko.dlut.edu.cn ESMTP Sendmail 8.9.1/8.9.1; Tue, 2 Feb 1999 10:41:20 +080
0 (CST)
HELP
214-This is Sendmail version 8.9.1
214-Topics:
214- HELO EHLO MAIL RCPT DATA
214- RSET NOOP QUIT HELP VRFY
214- EXPN VERB ETRN DSN
214-For more info use "HELP ".
214-To report bugs in the implementation send email to
214- sendmail-bugs@sendmail.org.
214-For local information send email to Postmaster at your site.
214 End of HELP info

以上目录结构是 Linux 的目录结构, Solaris 2.x 目录结构是:
/etc/mail/sendmail.cf
/etc/mail/sendmail.cw
/etc/mail/sendmail.hf
/etc/mail/aliases
/etc/mail/aliases.db

20. /etc/issue 系统进站提示信息(主控台用)

/etc/issue.net telnet 时显示信息(strings in.telnetd |egrep issue)
/etc/motd 用户进入系统后的提示信息

21. /etc/named.boot
DNS(BIND 4.9.x) 启动文件
示例文件:(Caching Only Server)
directory /etc/namedb

primary 0.0.127.in-addr.arpa named.local
cache . root.cache

其中 root.cache 文件可通过 dig 得到:

dig @ns.internic.net . ns > /etc/namedb/root.cache

named.local 文件内容如下:

@ IN SOA localhost. root.localhost. (
1999020301
10800
3600
86400
86400)

IN NS localhost.

1 IN PTR localhost.
/etc/named.conf
DNS(BIND 8.1.x) 启动文件
(在 Redhat 5.2 下可用/usr/doc/bind-8.1.2/named-bootconf.pl 将 bind 4.9.x
的 named.boot 文件转换为 bind8 的 named.conf 文件格式, 执行过程如下:
/usr/doc/bind-8.1.2/named-bootconf.pl /etc/named.boot > /etc/named.conf)

22. /etc/host.equiv
$HOME/.rhosts
R*(rlogin, rsh , rcp, rexec)服务信任主机
格式:
主机名(FQDN) 用户列表

23. /etc/ld.so.conf (LINUX)

动态链接库文件目录列表, 相应命令 ldconfig

$LD_LIBRARY_PATH Solaris 下相应的环境变量

用 ldd 列出相应文件所使用的动态链接库
/etc/default[119]ldd /usr/ucb/ls
libc.so.1 => /usr/lib/libc.so.1
libdl.so.1 => /usr/lib/libdl.so.1

*修改过此文件之后请使用命令 ldconfig 重新生成目录列表及连接库文件列表。

24. /etc/pam.d/login (Linux Redhat)
auth required /lib/security/pam_securetty.so
/etc/securetty (Linux Redhat, Debian)
root 可登录的终端设备列表, tty[1-8] 为主控台上的设备,
ttyp* (LINUX)远程登录终端(TELNET)设备

/etc/default/login (Solaris)
If CONSOLE is set, root can only login on that device.
Comment this line out to allow remote login by root.
#
CONSOLE=/dev/console
注释掉相应的记录即可允许超级用户 root 从远程主机 telnet 登录

/etc/login.defs Linux Debian 登录控制文件

25. Linux Loader /etc/lilo.conf
多重启动文件,
**** 修改完此文件后一定需要执行 lilo,
**** 重新编译安装新的 linux kernel 修改此文件并执行 lilo

26. /etc/syslog.conf
syslogd configuration file,

27. /etc/smb.conf
SAMBA 服务器配置文件,将 linux 的文件系统与 Windows 9x/NT 共享
28. /etc/nologin
系统在要关机时不希望用户登录进来，就产生此文件，此文件内容为显示给用户的有关拒绝连接的信息，用户此时就不能进入系统。当系统重新启动时如果有此文件，则机器启动后任何用户不能使用系统，此时可考虑从软盘或光盘引导删除此文件，然后

再重新启动系统。

29. /etc/security
设定那些终端可以让 root 登录，一般情况下设定为只有 console 上的用户可能用 root.

注：Redhat 下使用了 PAM 机制，相应的文件为/etc/securetty.

30. /etc/X11/*
XFree86 配置文件。

31. /etc/shells
用户可以使用的 shell 列表，如果强行修改/etc/passwd 文件，也可以使用不在列表中的 shell 程序，但对于 shell 不在此列表中的用户将无法使用 FTP 连接本系统。

32. /etc/mtab
系统在启动时创建的信息文件，内容为已经 mount 的文件系统，此文件内容是动态更新的，参考/proc/mounts。

197 ☆Linux 目录结构介绍

/:根目录，包含整个 LINUX 系统所有的目录和文件
/boot:系统启动时必须读取的文件，如系统内核，引导配置文件等
/dev:存放外围设备代号的文件，如硬盘的/dev/hda 它们实际上指向所代表的外围设备
/bin:这个目录放置运行时所使用的各种命令程序和不同的 shell
/etc:系统设置、管理相关的文件都放在这个目录下
/etc/rc.d:所有开关机所执行的脚本文件，都在这个目录下
/home:普通用户目录
/lib:包含许多被 /bin/ 和 /sbin/ 中的程序使用的库文件。目录 /usr/lib/ 中含有更多用于用户程序的库文件
/lib/modules:放置系统内核模块
/mnt:默认放置光盘和软盘的地方
/root:超级用户(管理员)的专用目录
/usr:包括与系统用户直接有关的文件和目录，例如应用程序及支持它们的库文件
/usr/bin: 里面是可以执行的命令程序，我们自己安装的软件，大多数可以在这里找到
/usr/src:存放 LINUX 源代码，内核源码也在这里面
/usr/X11R6 X 的文件目录，但配置文件不在这里，X 字体也在这里
/var:系统运行时的各种临时文件 log 文件，HTTPD 服务器和 FTPD 服务器等服务器的专用目录目录也在这里
/sbin:存放系统启动时所执行的程序
/tmp: 用户和程序的临时目录。 /tmp 给予所有系统用户读写权
/lost+found:被 fsck 用来放置零散文件（没有名称的文件）

/opt/ - 可选文件和程序的贮存目录。该目录主要被第三方开发者用来简易地安装和卸装他们的软件包
/proc:一个虚拟的文件系统（不是实际贮存在磁盘上的），它包括被某些程序使用的系统信息
/initrd/ - 用来在计算机启动时挂载 initrd.img 映像文件的目录以及载入所需设备模块的目录
警告： 不要删除 /initrd/ 目录。如果你删除了该目录后再重新引导 Linux 时，你将无法引导你的计算机。

198☆Linux 的基本配置文件

启动引导程序配置文件
LILO /etc/lilo.conf
GRUB /boot/grub/menu.lst

系统启动文件核脚本
主启动控制文件 /etc/inittab
SysV 启动脚本的位置 /etc/init.d、/etc/rc.d/init.d 或/etc/rc.d
SysV 启动脚本链接的位置 /etc/init.d/rc?.d、/etc/rc.d/rc?.d 或/etc/rc?.d
本地启动脚本 /etc/rc.d/rc.local、/etc/init.d/boot.local 或/etc/rc.boot 里的文件

网络配置文件
建立网络接口的脚本 /sbin/ifup
保存网络配置数据文件的目录 /etc/network、/etc/sysconfig/network 和/etc/sysconfig/network-scripts
保存解析 DNS 服务的文件 /etc/resolv.conf
DHCP 客户端的配置文件 /etc/dhclient.conf

超级服务程序配置文件和目录
inetd 配置文件 /etc/inetd.conf
TCP Wrappers 配置文件 /etc/hosts.allow 和/etc/hosts.deny
xinetd 配置文件 /etc/xinetd.conf 和/etc/xinetd.d 目录里的文件

硬件配置
内核模块配置文件 /etc/modules.conf

硬件访问文件
Linux 设备文件 /dev 目录里
保存硬件和驱动程序数据的文件 /proc 目录里

扫描仪配置文件
SANE 主配置 /etc/sane.d/dll.conf
特定扫描仪的配置文件 /etc/sane.d 目录里以扫描仪型号命名的文件

打印机配置文件

BSD LPD 核 LPRng 的本地打印机主配置文件 /etc/printcap
CUPS 本地打印机主配置和远程访问受权文件 /etc/cups/cupsd.conf
BSD LPD 远程访问受权文件 /etc/hosts.lpd
LPRng 远程访问受权文件 /etc/lpd.perms

文件系统
文件系统表 /etc/fstab
软驱装配点 /floppy、/mnt/floppy 或/media/floppy
光驱装配点 /cdrom、/mnt/cdrom 或/media/cdrom

shell 配置文件
bash 系统非登录配置文件 /etc/bashrc、/etc/bash.bashrc 或/etc/bash.bashrc.local
bash 系统登录文件 /etc/profile 和/etc/profile.d 里的文件
bash 用户非登录配置文件 ~/.bashrc
bash 用户登录配置文件 ~/.profile

XFree86 配置文件核目录
XFree86 主配置文件 /etc/XF86config、/etc/X11/XF86Config 或/etc/X11/XF86Config-4
字体服务程序配置文件 /etc/X11/fs/config
Xft 1.x 配置文件 /etcX11/XftConfig
Xft 2.0 配置文件 /etc/fonts/fonts.conf
字体目录 /usr/X11R6/lib/X11/fonts 和/usr/share/fonts

Web 服务程序配置文件
Apache 主配置文件 /etc/apache、/etc/httpd 或/httpd/conf 里的 httpd.conf 或 httpd2.conf 文件
MIME 类型文件 与 Apache 主配置文件在同一目录里的 mime.types 或 apache-mime.types

文件服务程序配置文件
ProFTPd 配置文件 /etc/proftpd.conf
vsftpd 配置文件 /etc/vsftpd.conf
NFS 服务程序的输出定义文件 /etc/exports
NFS 客户端装配的 NFS 输出 /etc/fstab
Samba 配置文件 /etc/samba/smb.conf
Samba 用户配置文件 /etc/samba/smbpasswd

邮件服务程序配置文件
sendmail 主配置文件 /etc/mail/sendmail.cf
sendmail 源配置文件 /etc/mail/sendmail.mc 或/usr/share/sendmail/cf/cf/linux.smtp.mc 或其他文件
Postfix 主配置文件 /etc/postfix/main.cf
Exim 主配置文件 /etc/exim/exim.cf
Procmail 配置文件 /etc/procmailrc 或~/.procmailrc
Fetchmail 配置文件 ~/.fetchmailrc

远程登录配置文件
SSH 服务程序配置文件 /etc/ssh/sshd_config
SSH 客户端配置文件 /etc/ssh/ssh_config
XDM 配置文件 /etc/X11/xdm 目录下
GDM 配置文件 /etc/X11/gdm 目录下
VNC 服务程序配置文件 /usr/X11R6/bin/vncserver 启动脚本和~/.vnc 目录里的文件

其他服务程序配置文件
DHCP 服务程序配置文件 /etc/dhcpd.conf
BIND 服务程序配置文件 /etc/named.conf 和/var/named/
NTP 服务程序配置文件 /etc/ntp.conf

199 ☆Linux 下的主要文件

/boot/grub/grub.conf GRUB configuration file
/boot/module-info-* Module information for the Linux kernel
/boot/System.map-* Map of the Linux kernel
/boot/vmlinuz-* Linux kernel
/etc/aliases Mail aliases
/etc/at.deny User IDs of users forbidden to use the at command
/etc/auto.master Configuration file for the autofs daemon, which automatically mounts filesystems
/etc/auto.misc Automounter map file
/etc/bashrc Systemwide functions and aliases for the bash shell
/etc/cron.daily/* Daily cron jobs
/etc/cron.hourly/* Hourly cron jobs
/etc/cron.monthly/* Monthly cron jobs
/etc/cron.weekly/* Weekly cron jobs
/etc/crontab System cron file
/etc/cups/* Printer configuration files
/etc/default/useradd Defaults for the useradd command
/etc/DIR_COLORS Directory listing colors
/etc/exports NFS exported directories
/etc/filesystems Supported filesystem types
/etc/fstab Filesystems mounted or available for mounting
/etc/group System group definitions
/etc/host.conf Resolver configuration file
/etc/hosts Map of IP numbers to hostnames
/etc/hosts.allow Hosts allowed to access Internet services
/etc/hosts.deny Hosts forbidden to access Internet services
/etc/httpd/conf/* Apache configuration files

/etc/httpd/httpd.conf Main Apache configuration file
/etc/init.d/* SysV initialization scripts
/etc/initlog.conf Logging configuration file
/etc/inittab Configuration for the init daemon, which controls executing processes
/etc/issue Linux kernel and distribution version (local users)
/etc/issue/net Linux kernel and distribution version (remote users)
/etc/ld.so.conf Shared library configuration file
/etc/login.defs Options for useradd and related commands
/etc/logrotate.conf Log rotation configuration file
/etc/logrotate.d/* Scripts to rotate logs
/etc/mail/* Mail server configuration files
/etc/mailcap metamail MIME information
/etc/man.config man configuration file
/etc/mime.types MIME types
/etc/mime-magic* Magic numbers for MIME data
/etc/minicom.users User IDs allowed to use minicom
/etc/modules.conf Aliases and options for loadable kernel modules
/etc/motd Message of the day
/etc/mtab Mounted filesystems
/etc/nsswitch.conf Resolver configuration file
/etc/openldap/* Open LDAP configuration files
/etc/pam.d/* PAM configuration files
/etc/paper.config Paper sizes
/etc/passwd User account information
/etc/ppp/* PPP configuration
/etc/printcap Printer options and capabilities
/etc/profile Default environment for users of the bash shell
/etc/profile.d/* Shell initialization
/etc/protocols Protocol names and numbers
/etc/pwdb.conf pwdb library configuration
/etc/rc Scripts for system and process startup and shutdown
/etc/rc.local Local startup script
/etc/rc.sysinit System initialization file
/etc/rc?.d/* Service start/stop scripts
/etc/rpc RPC program number database
/etc/rpm/* RPM database and configuration files
/etc/samba/* Samba configuration files
/etc/securetty Secure tty configuration
/etc/security/* PAM configuration files
/etc/sensors.conf libsensors configuration file
/etc/services Standard service names and numbers
/etc/shadow Secure user account information

/etc/skel Skeleton files used to establish new user accounts
/etc/ssh/* SSH configuration files
/etc/sysconfig/* System configuration files
/etc/sysconfig/network-scripts/* Network adapter configuration files
/etc/sysctl.conf sysctl configuration file
/etc/syslog.conf System logging process configuration
/etc/termcap Terminal capabilities and options
/etc/updatedb.conf updatedb/locate configuration file
/etc/wvdial.conf GNOME dialer configuration file
/etc/X11/applnk/* X application shortcuts
/etc/X11/fs/config X font server configuration
/etc/X11/gdm/* GNOME display manager configuration
/etc/X11/prefdm Display manager configuration file
/etc/X11/xdm/* X display manager configuration file
/etc/X11/XF86Config X configuration file
/etc/X11/xinit/Xclients Default script for xinit
/etc/X11/xinit/xinitrc X session initialization file
/etc/X11/Xmodmap Key mappings used by xdm and xinit
/etc/xinetd.conf General xinetd configuration file
/etc/xinetd.d/* xinetd configuration files for specific servers
/home/*/public_html User web pages
/root/.bash_history bash command history for system administrator
/root/.bash_logout bash logout script for system administrator
/root/.bash_profile bash initialization script for system administrator
/root/.bashrc bash options for system administrator
/root/.Xresources X resources for system administrator
/usr/share/config/* Miscellaneous configuration files
/usr/share/fonts/* Fonts
/usr/share/ssl/openssl.cnf SSL certificate configuration
/usr/X11R6/lib/X11/app-defaults/* X application defaults
/usr/X11R6/lib/X11/fonts/* X fonts
/var/log/cron Log of cron activity
/var/log/httpd/access_log Log of web server access
/var/log/httpd/error_log Log of web server errors
/var/log/boot.log Boot messages
/var/log/cron Cron log
/var/log/dmesg Kernel message log
/var/log/lastlog Last login log
/var/log/maillog Mail transfer log
/var/log/messages System log
/var/log/samba/* Samba logs
/var/log/secure System security log

/var/log/up2date Up2date log
/var/www/cgi-bin CGI scripts
/var/www/html/* Web pages

200 ☆linux 路径

Linux 文件系统是从/开始的
绝对路径是从/（也被称为根目录）开始的
相对路径是以 . 或 .. 开始的
 .表示用户当前操作所处的位置，
而.. 表示上级目录;
在路径中，.表示用户当前所处的目录，而..表示上级目录，
要把.和..当做目录来看。

 . 表示用户所处的当前目录;
 .. 表示上级目录
~ 表示当前用户自己的家目录
~USER 表示用户名为 USER 的家目录，这里的 USER 是在/etc/passwd 中存在的用户名;

201 ☆虚拟控制台

linux 允许用户在同一时间从控制台（系统的控制台是与系统直接相连的监视器和键盘）进行多次登录。
虚拟控制台的选择可以通过按下 Alt 键和一个功能键来实现，通常使用 F1-F6。
ALT+SHIFT+F7 是返回图形桌面;

202 ☆type

type [命令|文件|别名]

[root@panda ~]# type set
set is a shell builtin

[root@panda ~]# type while
while is a shell keyword

[root@panda ~]# type file
file is /usr/bin/file

[root@panda ~]# type ll
ll is aliased to `ls -l --color=tty'

203 ☆whereis

语法: whereis 命令
显示命令的路径
寻找命令的二进制文件，同时也会找到其帮助文件;

[root@panda ~]# whereis ls
ls: /bin/ls /usr/share/man/man1p/ls.1p.gz /usr/share/man/man1/ls.1.gz

204 ☆which

语法: which [命令|别名|文件]

显示命令的路径,及使用者所定义的别名.
只从我们所设置的环境变量中设置好的路径中寻找;

参数:
-c 打印输出幻数文件的分析形式。这通常用于与-m 联合使用。在安装一个新的幻数文件之前调试它
-z 深入观察一个压缩文件，并试图查出他的类型
-f ffile：告诉 file 要鉴别的文件列表在 ffile 中。这对于需要鉴别许多文件很有用
-L 本选项允许符号连接
-m 文件：指定用于说明文件类型的幻数的一个替换文件

[root@panda ~]# which httpd
/usr/sbin/httpd

[root@panda ~]# which file
/usr/bin/file

[root@panda ~]# which ls
alias ls='ls --color=tty'
/bin/ls

205 ☆info

格式:
info 查询命令

作用：交互式文档工具

206 ☆whatis

语法: whatis 命令
显示命令(man 中)功能的摘要.

[root@panda ~]# whatis ls
ls (1) - list directory contents

207 ☆apropos

显示 man 中关于命令的所有页

[root@panda temp]$apropos mknod
mknod (1) - make block or character special files
mknod (2) - create a special or ordinary file
vgmknodes (8) - recreate volume group directory and logical volume special files

208 ☆man

语法:
man 命令

如:man ls

209 ☆pwd

语法:
pwd

说明:
此命令显示出当前工作目录的绝对路径。

210 ☆ls

(list)
使用权限:所有使用者

语法:
ls [选项] [目录或是文件]

说明:
无参数的时候,显示指定工作目录下之内容(列出目前工作目录所含之文件及子目录)。

对于每个目录，该命令将列出其中的所有子目录与文件。
对于每个文件，ls 将输出其文件名以及所要求的其他信息。
默认情况下，输出条目按字母顺序排序。当未给出目录名或是文件名时，就显示当前目录的信息。

参数:
-a 显示指定目录下所有子目录与文件，包括隐藏文件。(ls 内定将文件名或目录名称开头为"."的视为隐藏档，不会列出)
-A 同-a，但不列出"."(目前目录)及".."(父目录)
-l 以长格式来显示文件的详细信息。除文件名称外，亦将文件型态、权限、拥有者、文件大小等资讯详细列出
-n 输出格式与 l 选项相同，只不过在输出中文件属主和属组是用相应的 UID 号和 GID 号来表示，而不是实际的名称。
-o 与 l 选项相同，只是不显示拥有组信息。
-i 在输出的第一列显示文件的 i 节点号。
-s 给出每个目录项所用的块数，包括间接块。
-L 若指定的名称为一个符号链接文件，则显示链接所指向的文件。
-d 如果参数是目录，只显示其名称而不显示其下的各文件。往往与 l 选项一起使用，以得到目录的详细信息。
-f 不排序。该选项将使 lts 选项失效，并使 aU 选项有效。
-r 将文件以相反次序显示(原定依英文字母次序),按字母逆序或最早优先的顺序显示输出结果。
-t 显示时按修改时间(最近优先)而不是按名字排序。若文件修改时间相同，则按字典顺序。修改时间取决于是否使用了 c 或 u 选项。缺省的时间标记是最后一次修改时间。将文件依建立时间之先后次序列出
-u 显示时按文件上次访问(存取)的时间(最近优先)而不是按名字排序。即将-t 的时间标记修改为最后一次访问的时间。
-c 按文件的修改时间排序。
-F 在列出的文件名称后加一符号;在目录名后面标记"/"，可执行文件后面标记"*"，符号链接后面标记"@"，管道(或 FIFO)后面标记"|"，socket 文件后面标记"="。
-p 在目录后面加一个"/"。
-R 递归式地显示指定目录的各个子目录中的文件。若目录下有文件，则以下之文件亦皆依序列出
-q 将文件名中的不可显示字符用"?"代替。
-b 对文件名中的不可显示字符用八进制逃逸字符显示。
-C 分成多列显示各项。(缺省)
-x 按行显示出各排序项的信息。行的横向排序,缺省是按列的竖向排序
-m 输出按字符流格式，文件跨页显示，以逗号分开。
-B 不列出文件名以~结尾的文件;

在文件的大小排序上，我们要用到-S 参数;
如果是逆序排序时，我们要用到-r 参数;
按最后访问的时候排序，要用到-t 参数;
根据扩展名进行排序，要用到参数 -X ;

--color=never 表示输出输出没有彩色
--color=auto 表示自动
--color=always 表示输出内容有彩色

该命令可以使用通配符,查看通配符匹配范围
* 代表 0 个或多个字符
[]内部包括任何字符
? 任何单个字符

ls -l *dec?b?

ls -1 /usr/bin | wc -l

211 ☆du

du 的英文原义为"disk usage"，含义为显示磁盘空间的使用情况。

功能:统计目录(或文件)所占磁盘空间的大小。

语法:du [选项] [Names…]

说明:该命令逐级进入指定目录的每一个子目录并显示该目录占用文件系统数据块(1024 字节)的情况。若没有给出 Names，则对当前目录进行统计。

该命令的各个选项含义如下:

-h 以更可读的方法输出
-s 对每个 Names 参数只给出占用的数据块总数。
-a 递归地显示指定目录中各文件及子目录中各文件占用的数据块数。若既不指定-s，也不指定-a，则只显示 Names 中的每一个目录及其中的各子目录所占的磁盘块数。
-b 以字节为单位列出磁盘空间使用情况(系统缺省以 k 字节为单位)。
-k 以 1024 字节为单位列出磁盘空间使用情况。
-c 最后再加上一个总计(系统缺省设置)。
-l 计算所有的文件大小，对硬链接文件，则计算多次。
-x 跳过在不同文件系统上的目录不予统计。

下面举例说明 du 命令的使用:

列出各目录所占的磁盘空间，但不详细列出每个文件所占的空间。
$ du
4 ./november

```
4       ./a_reports/2
4       ./a_reports/1/temp/2/1
8       ./a_reports/1/temp/2
4       ./a_reports/1/temp/1
16      ./a_reports/1/temp
20      ./a_reports/1
4       ./a_reports/3
32      ./a_reports
8       ./.xemacs
4       ./dir1
4       ./october
4       ./december
4       ./september
88      .
```
输出清单中的第一列是以块为单位计的磁盘空间容量，第二列列出目录中使用这些空间的目录名称。

注意不带选项的 du 命令将从当前目录开始沿着目录结构向下工作直到列出所有目录的容量为止。这可能是一个很长的清单，有时只需要一个总数。这时可在 du 命令中加-s 选项来取得总数:

列出所有文件和目录所占的空间(使用 a 选项)，而且以字节为单位(使用 b 选项)来计算大小。
$ du -ab

du dir1
显示目录 dir1 的总容量及其子目录的容量(以 KB 为单位)

du -s dir1
显示目录 dir1 的总容量

du -sh
查看文件夹大小

212☆mkdir

功能:
创建一个目录

语法:
mkdir [选项] dirname

说明:
该命令创建由 dirname 命名的目录。

要求创建目录的用户在当前目录中(dirname 的父目录中)具有写权限
并且 dirname 不能是当前目录中已有的目录或文件名称。

命令中各选项的含义为:
-m 对新建目录设置存取权限。也可以用 chmod 命令设置。
-p 可以是一个路径名称。此时若路径中的某些目录尚不存在，加上此选项后，系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录。

在当前目录中建立 panda 和 panda 下的/mail 目录，也就是连续建两个目录。
$ mkdir -p -m 700 ./panda/mail/

213 ☆rmdir

使用权限:于目前目录有适当权限的所有使用者

语法:
rmdir [选项] dirName

说明:
删除空的目录。
dirName 表示目录名。该命令从一个目录中删除一个或多个子目录项。需要特别注意的是，一个目录被删除之前必须是空的。(注意，rm -r dir 命令可代替 rmdir，但是有很大危险性。)删除某目录时也必须具有对父目录的写权限。

参数:
-p 递归删除目录 dirName，当子目录删除后其父目录为空时，也一同被删除。如果整个路径被删除或者由于某种原因保留部分路径，则系统在标准输出上显示相应的信息。

将工作目录下，名为 dir1 的子目录删除 :
rmdir dir1

rmdir -p dir1/Test
在工作目录下的 dir1 目录中，删除名为 Test 的子目录。若 Test 删除后，dir1 目录成为空目录，则 dir1 亦予删除。必须写全路径,不能只写最上层,否则认为是非空目录

rmdir dir1/Test
只删除 Test 目录,没有-p 就只删除最低级目录

214 ☆cd

使用权限:
所有使用者

语法:
cd [dirName]

说明:
变换工作目录至 dirName。
为了改变到指定目录，用户必须拥有对指定目录的执行和读权限。
其中 dirName 表示法可为绝对路径或相对路径。
若目录名称省略，则变换至使用者的 home directory(也就是刚 login 时所在的目录)。

另外，"~"也表示为 home directory 的意思
"."则是表示目前所在的目录
".."则表示目前目录位置的上一层目录

该命令可以使用通配符

跳到 /usr/bin/
cd /usr/bin

假设用户当前目录是:/home/xu，现需要更换到/home/xu/pro 目录中，
$ cd pro

跳到自己的 home directory :
cd ~

跳到目前目录的上上两层 :
cd ../..

215 ☆touch

使用权限:
所有使用者

使用方式:
touch [-acfm]
[-r reference-file] [--file=reference-file]
[-t MMDDhhmm[[CC]YY][.ss]]
[-d time] [--date=time] [--time={atime,access,use,mtime,modify}]
[--no-create] [--help] [--version]
file1 [file2 ...]

说明:

touch 指令改变文件的时间记录。
ls -l 可以显示文件的时间记录。

参数:
-a 改变文件的读取时间记录。
-m 改变文件的修改时间记录。
-d 设定时间与日期，可以使用各种不同的格式。
-t 设定文件的时间记录，格式与 date 指令相同。
-c 假如目的文件不存在，不会建立新的文件。与 --no-create 的效果一样。
-f 不使用，是为了与其他 unix 系统的相容性而保留。
-r 使用参考档的时间记录，与 --file 的效果一样。
--no-create 不会建立新文件。
--help 列出指令格式。
--version 列出版本讯息。

最简单的使用方式，将文件的时候记录改为现在的时间。若文件不存在，系统会建立一个新的文件。
touch file
touch file1 file2

将 file 的时间记录改为 5 月 6 日 18 点 3 分，公元两千年。时间的格式可以参考 date 指令，至少需输入 MMDDHHmm ，就是月日时与分。
touch -c -t 05061803 file
touch -c -t 050618032000 file

将 file 的时间记录改变成与 referencefile 一样。
touch -r referencefile file

将 file 的时间记录改成 5 月 6 日 18 点 3 分，公元两千年。时间可以使用 am, pm 或是 24 小时的格式，日期可以使用其他格式如 6 May 2000 。
touch -d "6:03pm" file
touch -d "05/06/2000" file
touch -d "6:03pm 05/06/2000" file

touch {report,memo,graph}_{sep,oct,nov,dec}_{a,b,c}{1,2,3}

216☆cp

(copy)
使用权限:所有使用者

语法:

cp [选项] 源文件或目录... 目标文件或目录

说明:
该命令把指定的源文件复制到目标文件或把多个源文件复制到目标目录中。

参数
-a 该选项通常在拷贝目录时使用。它保留链接、文件属性，并递归地拷贝目录，其作用等于 dpR 选项的组合。尽可能将文件状态、权限等资料都照原状予以复制。
-p 此时 cp 除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中。
-P 通过加入目标目录分支和指定的原文件名形成每个目标文件名。给 cp 的最后一个变量必须是已存在的目录的名字。
-r 若给出的源中含有目录文件，此时 cp 将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名。如果源中有目录,没有加-r 参数的时候会 cp: omitting directory
-f 删除已经存在的目标文件而不提示。若目的地已经有相同档名的文件存在，则在复制前先予以删除再行复制。
-i 和 f 选项相反，在覆盖目标文件之前将给出提示要求用户确认。回答 y 时目标文件将被覆盖，是交互式拷贝。
-d 不间接引用符号链接，保持源文件和目标文件之间的硬链接关系;拷贝时保留链接。将符号连接作为符号连接拷贝，而不拷贝他们所指向的文件。并在备份中保持源文件间固有的链接关系
-l 不作拷贝，只是链接文件。形成硬链接以代替非目录的拷贝
-b 做将要覆盖或删除文件的备份,注意文件名后有~结尾
-u 自动更新，只靠被改变过的文件
-R 递规拷贝目录
-v 在拷贝前打印每个文件名
-s 建立符号连接，而不是复制源文件。源文件名必须用绝对路径;如果目标文件不在当前目录，所有的原文件名必须是绝对路径(从‘/’开始)。对不支持符号连接的系统，本选项将产生一个错误信息
-d 将符号连接作为符号连接拷贝，而不拷贝他们所指向的文件。并在备份中保持源文件间固有的链接关系

将文件 file1 复制(已存在)，并命名为 file2 :
cp file1 file2

将所有的 C 语言程式拷贝至 temp 子目录中 :
cp *.c temp

需要说明的是，为防止用户在不经意的情况下用 cp 命令破坏另一个文件，如用户指定的目标文件名是一个已存在的文件名，用 cp 命令拷贝文件后，这个文件就会被新拷贝的源文件覆盖，因此，建议用户在使用 cp 命令拷贝文件时，最好使用-i 选项。

$ cp -i exam1.c /usr/wang/shiyan1.c

该命令将文件 exam1.c 拷贝到/usr/wang 这个目录下，并改名为 shiyan1.c。若不希望重新命名，可以使用下面的命令:
$ cp exam1.c /usr/wang/

$ cp -r /usr/xu/ /usr/liu/
将/usr/xu 目录中的所有文件及其子目录拷贝到目录/usr/liu 中。

217 ☆mv

(move)
使用权限:所有使用者

语法:
mv [选项] 源文件或目录... 目标文件或目录

说明:
用户可以使用 mv 命令来为文件或目录改名或将文件由一个目录移入另一个目录中。
视 mv 命令中第二个参数类型的不同(是目标文件还是目标目录)，mv 命令将文件重命名或将其移至一个新的目录中。
当第二个参数类型是文件时，mv 命令完成文件重命名，此时，源文件只能有一个(也可以是源目录名)，它将所给的源文件或目录重命名为给定的目标文件名。
当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，mv 命令将各参数指定的源文件均移至目标目录中。
在跨文件系统移动文件时，mv 先拷贝，再将原有文件删除，而链至该文件的链接也将丢失。

参数:
-i 交互方式操作。如果 mv 操作将导致对已存在的目标文件的覆盖，此时系统询问是否重写，要求用户回答 y 或 n，这样可以避免误覆盖文件。
-f 禁止交互操作。在 mv 操作要覆盖某已有的目标文件时不给任何指示，指定此选项后，i 选项将不再起作用。
-b 为要移动的文件制作备份
-u 在目标文件的时间原文件新时不覆盖目标文件

如果所给目标文件(不是目录)已存在，此时该文件的内容将被新文件覆盖。为防止用户在不经意的情况下用 mv 命令破坏另一个文件，建议用户在使用 mv 命令移动文件时，最好使用 i 选项。

需要注意的是，mv 与 cp 的结果不同。mv 好象文件"搬家"，文件个数并未增加，而 cp 对文件进行复制，文件个数增加了。

将/usr/panda 中的所有文件移到当前目录(用"."表示)中:
$ mv /usr/panda/* .

将文件 panda.txt 重命名为 pandanew.doc
$ mv panda.txt pandanew.doc

将所有的 C 语言程式移至 temp 子目录中 :
mv -i *.c temp

218 ☆rm

(remove)
使用权限:所有使用者

语法:
rm [选项] 文件…

说明:
该命令的功能为删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。
如果没有使用-r 选项，则 rm 不会删除目录。

参数:
-i 进行交互式删除。删除前逐一询问确认。
-f 忽略不存在的文件，从不给出提示。即使原文件属性设为唯读，亦直接删除，无需逐一确认。
-r 将目录及以下之文件亦逐一递归地删除。
-v 删除每个文件时输出文件信息

使用 rm 命令要格外小心。因为一旦一个文件被删除，它是不能被恢复的。例如，用户在输入 cp，mv 或其他命令时，不小心误输入了 rm 命令，当用户按了回车键并认识到自己的错误时，已经太晚了，文件已经没有了。为了防止此种情况的发生，可以使用 rm 命令中的-i 选项来确认要删除的每个文件。如果用户输入 y，文件将被删除。如果输入任何其他东西，文件将被保留。在下一个例子中，用户要删除文件 file1 和 file2。然后会被要求对每个文件进行确认。用户最终决定删除 file2 文件，保留 file1 文件。

$ rm -i file1 file2
Remove file1 ?n
Remove file2 ?y

删除所有 C 语言程式档;删除前逐一询问确认 :
rm -i *.c

将 panda 子目录及子目录中所有文件删除 :
rm -r panda

219 ☆ln

使用权限:所有使用者

语法:
ln [选项] 目标 [链接名]
ln [选项] 目标 目录

其中[选项]的格式为 :
[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}]
[--help] [--version] [--]

说明:
该命令在文件之间创建链接。这种操作实际上是给系统中已有的某个文件指定另外一个可用于访问它的名称。对于这个新的文件名，我们可以为之指定不同的访问权限，以控制对信息的共享和安全性的问题。
如果链接指向目录，用户就可以利用该链接直接进入被链接的目录而不用打一大堆的路径名。而且，即使我们删除这个链接，也不会破坏原来的目录。

链接有两种，一种被称为硬链接(Hard Link)，另一种被称为符号链接(Symbolic Link)。
硬连结的意思是一个文件可以有多个名称，而软连结的方式则是产生一个特殊的文件，该文件的内容是指向另一个文件的位置。硬连结是存在同一个文件系统中，而软连结却可以跨越不同的文件系统。
建立硬链接时，链接文件和被链接文件必须位于同一个文件系统中，并且不能建立指向目录的硬链接。
而对符号链接，则不存在这个问题。
不论是硬连结或软链结都不会将原本的文件复制一份，只会占用非常少量的磁碟空间。
默认情况下，ln 产生硬链接。

在硬链接的情况下，参数中的"目标"被链接至[链接名]。
如果[链接名]是一个目录名，系统将在该目录之下建立一个或多个与"目标"同名的链接文件，链接文件和被链接文件的内容完全相同。
如果[链接名]为一个文件，用户将被告知该文件已存在且不进行链接。如果指定了多个"目标"参数，那么最后一个参数必须为目录。

如果给 ln 命令加上-s 选项，则建立符号链接。
如果[链接名]已经存在但不是目录，将不做链接。[链接名]可以是任何一个文件名(可包含路径)，也可以是一个目录，并且允许它与"目标"不在同一个文件系统中。
如果[链接名]是一个已经存在的目录，系统将在该目录下建立一个或多个与"目标"同名的文件，此新建的文件实际上是指向原"目标"的符号链接文件。

inode 译成中文就是索引节点。每个存储设备或存储设备的分区被格式化为文件系统后，应该有两部份，一部份是 inode，另一部份是 Block，Block 是用来存储数据用的。而 inode 呢，就是用来存储这些数据的信息，这些信息包括文件大小、属主、归属的用户组、读写权限等。inode 为每个文件进行信息索引，所以就有了 inode 的数值。操作系统根据指令，能通过 inode 值最快的找到相对应的文件。

用 ln 创建文件硬链接的语法：
ln 源文件 目标文件

inode 相同的文件是硬链接文件;inode 值相同的文件，他们的关系是互为硬链接的关系。不同的文件名，inode 可能是相同的，一个 inode 值可以对应多个文件。当我们修改其中一个文件的内容时，互为硬链接的文件的内容也会跟着变化。如果我们删除互为硬链接关系的某个文件时，其它的文件并不受影响。

可以这么理解，互为硬链接关系的文件，他们好象是克隆体，他们的属性几乎是完全一样;

对软链接生成硬链接,结果还是一个软接连,会指向原文件

注意：硬链接不能为目录创建，只有文件才能创建硬链接。

创建软链接（也被称为符号链接）的语法;
ln -s 源文文件或目录 目标文件或目录

软链接也叫符号链接，他和硬链接有所不同，软链接文件只是其源文件的一个标记。当我们删除了源文件后，链接文件不能独立存在，虽然仍保留文件名，但我们却不能查看软链接文件的内容了。

值得我们注意的是：当我们修改链接文件的内容时，就意味着我们在修改源文件的内容。当然源文件的属性也会发生改变，链接文件的属性并不会发生变化。当我们把源文件删除后，链接文件只存在一个文件名，因为失去了源文件，所以软链接文件也就不存在了。这一点和硬链接是不同的;

我们可以看到软链接文件，其实只是源文件的一个标记，当源文件失去时，他也就是存在了。软链接文件只是占用了 inode 来存储软链接文件属性等信息，但文件存储是指向源文件的。

软件链接，可以为文件或目录都适用。无论是软链接还是硬链接，都可以用 rm 来删除。rm 工具是通用的。

参数:
-f : 链结时先将与 ln 的目标的同名的文件删除
-d : 允许系统管理者硬链结自己的目录
-i : 在删除与 dist 同档名的文件时先进行询问

-n : 在进行软连结时，将 dist 视为一般的文件
-s : 进行软链结(symbolic link)
-v : 在连结之前显示其档名
-b : 将在链结时会被覆写或删除的文件进行备份
-S SUFFIX : 将备份的文件都加上 SUFFIX 的字尾
-V METHOD : 指定备份的方式
--help : 显示辅助说明
--version : 显示版本

将文件 yy 产生一个 symbolic link : zz
ln -s yy zz

将文件 yy 产生一个 hard link : xx
ln yy xx

可以创建一个目标文件不存在的软链接
ln -s /panda/panda test01

当我们用 ls 查看某个目录或文件时，如果加上-i 参数，就可以看到 inode 节点了;比如我们前面所说的例子;
[root@localhost ~]# ls -li lsfile.sh
2408949 -rwxr-xr-x 1 root root 7 04-21 12:47 lsfile.sh

220☆chmod

使用权限:所有使用者

语法:
chmod [-cfvR] [--help] [--version] mode file...

说明:
Linux/Unix 的文件存取权限分为三级 : 文件拥有者、群组、其他。利用 chmod 可以藉以控制文件如何被他人所存取。
chmod 是用来改变文件或目录权限的命令，但只有文件的属主和超级权限用户 root 才有这种权限。通过

参数:
mode : 权限设定字串，格式如下 : [ugoa...][[+-=][rwxX]...][,...]，其中 u 表示该文件的拥有者，g 表示与该文件的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。
+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。
r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该文件是个子目录或者该文件已经被设定过为可执行。

-c : 若该文件权限确实已经更改，才显示其更改动作
-f : 若该文件权限无法被更改也不要显示错误讯息
-v : 显示权限变更的详细资料
-R : 对目前目录下的所有文件与子目录进行相同的权限变更(即以递回的方式逐个变更)
--help : 显示辅助说明
--version : 显示版本

改变文件或目录的权限有两种方法。一种是包含字母和操作符表达式的文字设定法;另一种是包含数字的数字设定法。

1.文字设定法
chmod [who] [+|-|=] [mode] 文件名

命令中各选项的含义为:
操作对象 who 可是下述字母中的任一个或者它们的组合:
u 表示"用户(user)",即文件或目录的所有者.
g 表示"同组(group)用户",即与文件属主有相同组 ID 的所有用户.
o 表示"其他(others)用户".
a 表示"所有(all)用户".它是系统默认值.

操作符号可以是:
+ 添加某个权限.
- 取消某个权限.
= 赋予给定权限并取消其他所有权限(如果有的话).

设置 mode 所表示的权限可用下述字母的任意组合:
r 可读.
w 可写.
x 可执行.
X 只有目标文件对某些用户是可执行的或该目标文件是目录时才追加 x 属性.
s 在文件执行时把进程的属主或组 ID 置为该文件的文件属主.方式"u+s"设置文件的用户 ID 位,"g+s"设置组 ID 位.
t 保存程序的文本到交换设备上.
u 与文件属主拥有一样的权限.
g 与和文件属主同组的用户拥有一样的权限.
o 与其他用户拥有一样的权限.

通过 u+s 或 u-s 来增减 setuid 位
通过 g+s 或 g-s 来增减 setgid 位
通过 o+t 或 o-t 来增减粘帖位

文件名:以空格分开的要改变权限的文件列表,支持通配符.

在一个命令行中可给出多个权限方式,其间用逗号隔开.

chmod g+r,o+r example

chmod ug+w,o-x text

$ chmod a-x mm.txt
$ chmod -x mm.txt
$ chmod ugo-x mm.txt
以上这三个命令都是将文件 mm.txt 的执行权限删除,它设定的对象为所有使用者.

2. 数字设定法
我们必须首先了解用数字表示的属性的含义:0 表示没有权限,1 表示可执行权限,2 表示可写权限,4 表示可读权限,然后将其相加.所以数字属性的格式应为 3 个从 0 到 7 的八进制数,其顺序是(u)(g)(o).

数字设定法的一般形式为:
chmod [mode] 文件名

chmod 的八进制语法的数字说明;
r: 对应数值 4
w: 对应数值 2
x：对应数值 1
-：对应数值 0

如果要加上特殊权限，就必须使用 4 位数字才能表示。特殊权限的对应数值为：
s 或 S （SUID）：对应数值 4。
s 或 S （SGID）：对应数值 2。
t 或 T ：对应数值 1。

setuid 位是的设置用八进制的 4000，setgid 占用的是八进制的 2000 ;粘贴位的设置，可以用八进制的 1000 位来设置

$ chmod 644 mm.txt

$ chmod 750 wch.txt

221☆chown

使用权限:root

语法:

chown [选项]... [所有者][:[组]] 文件...
chmod [-cfhvR] [--help] [--version] user[:group] file...

说明 :
当我们要改变一个文件的属组，我们所使用的用户必须是该文件的属主而且同时是目标属组成员，或超级用户。只有超级用户的才能改变文件的属主。

chown 所接的新的属主和新的属组之间应该以.或:连接，属主和属组之一可以为空。如果属主为空，应该是 :属组 ;如果属组为空 就就不必需要.或:了。

chown 将指定文件的拥有者改为指定的用户或组。用户可以是用户名或用户 ID。组可以是组名或组 ID。文件是以空格分开的要改变权限的文件列表，支持通配符。

参数:
user : 新的文件拥有者的使用者 ID
group : 新的文件拥有者的使用者群体(group)
-c : 若该文件拥有者确实已经更改，才显示其更改动作
-f : 若该文件拥有者无法被更改也不要显示错误讯息
-h : 只对于连结(link)进行变更，而非该 link 真正指向的文件
-v : 显示 chown 命令所做的工作。
-R : 递归式地改变指定目录及其下的所有子目录和文件的拥有者。
--help : 显示辅助说明
--version : 显示版本

把文件 file.c 的所有者改为 panda。
$ chown panda file.c

把目录/his 及其下的所有文件和子目录的属主改成 wang，属组改成 users。
$ chown -R panda.root /temp

将文件 file1.txt 的拥有者设为 users 群体的使用者 panda :
chown panda:users file1.txt

将目前目录下的所有文件与子目录的拥有者皆设为 users 群体的使用者 panda :
chmod -R panda:users *

要改变所属组，可使用下面命令：
[root@localhost ~]# chown :users panda.file

222☆chgrp

语法:

chgrp [选项] group filename

说明:
改变文件或目录所属的组。
该命令改变指定指定文件所属的用户组。其中 group 可以是用户组 ID，也可以是/etc/group 文件中用户组的组名。文件名是以空格分开的要改变属组的文件列表，支持通配符。如果用户不是该文件的属主或超级用户，则不能改变该文件的组。

参数:
-R 递归式地改变指定目录及其下的所有子目录和文件的属组。

$ chgrp -R users /opt/local/panda
改变/opt/local/panda/及其子目录下的所有文件的属组为 book。

223☆umask

umask 是通过八进制的数值来定义用户创建文件或目录的默认权限。
umask 表示的是禁止权限。不过文件和目录有点不同。

对于文件来说，umask 的设置是在假定文件拥有八进制 666 权限上进行，文件的权限就是是 666 减去 umask 的掩码数值;
对于目录来说，umask 的设置是在假定文件拥有八进制 777 权限上进行，目录八进制权限 777 减去 umask 的掩码数值;

umask 一般都是放在用户相关 SHELL 的配置文件中，比如用户家目录下的.bashrc 或.profile ，也可以放在全局性的用户配置文件中，比如 /etc/login.defs，还可以放在 SHELL 全局的配置文件中，比如/etc/profile 或/etc/bashrc 或/etc/csh.cshrc 等;

umask 放在相关的配置文件中，目的是当管理员创建用户时，系统会自动为用户创建文件或目录时配置默认的权限代码。

224☆lsattr

参数:
-a
列出目录中的所有文件，包括以.开头的文件。

-d
以和文件相同的方式列出目录，并显示其包含的内容。

-R
以递归的方式列出目录的属性及其内容。

-v
列出文件版本(用于网络文件系统 NFS)。

225☆chattr

ext2 文件系统就开始支持一些针对文件和目录的额外标记或者叫作属性(attribute)。

修改 ext2 和 ext3 文件系统属性(attribute)，使用权限超级用户。

格式
chattr [-RV] [-+=AacDdijsSu] [-v version] 文件或目录

参数
-R：递归处理所有的文件及子目录。
-V：详细显示修改内容，并打印输出。

-：失效属性。
+：激活属性。
=：指定属性。

A：Atime，告诉系统不要修改对这个文件的最后访问时间。
S：Sync，一旦应用程序对这个文件执行了写操作，使系统立刻把修改的结果写到磁盘。
a：Append Only，系统只允许在这个文件之后追加数据，不允许任何进程覆盖或截断这个文件。如果目录具有这个属性，系统将只允许在这个目录下建立和修改文件，而不允许删除任何文件。
i：Immutable，系统不允许对这个文件进行任何的修改。如果目录具有这个属性，那么任何的进程只能修改目录之下的文件，不允许建立和删除文件。
D：检查压缩文件中的错误。
d：No dump，在进行文件系统备份时，dump 程序将忽略这个文件。
C：Compress，系统以透明的方式压缩这个文件。从这个文件读取时，返回的是解压之后的数据;而向这个文件中写入数据时，数据首先被压缩之后才写入磁盘。
s：Secure Delete，让系统在删除这个文件时，使用 0 填充文件所在的区域。
u：Undelete，当一个应用程序请求删除这个文件，系统会保留其数据块以便以后能够恢复删除这个文件。

说明:
chattr 命令不能保护/、/dev、/tmp、/var 目录。
不能使用 chattr 命令的目录
/
很显然，根分区不能有 immutable 属性。如果根分区具有 immutable 属性，系统将根本无法工作。
/dev

在启动时，syslog 需要删除并重新建立/dev/log 套接字设备。如果对/dev/目录设置了 immutable 和 append-only 属性，就可能出现问题，除 非在启动 syslogd 时使用-p 选项指定其它的套接字，例如：/var/run/syslog.sock。即使这样也还存在一些问题，syslog 客户程序需要/dev/log 套接字设备，因此需要建立一个自相真正套接字的符号连接。总而言之，为了减少麻烦，这个目录还是不要设置 immutable 和 append-only 属性。
/tmp
有很多应用程序和系统程序需要在这个目录下建立临时文件，因此这个目录也不能设置 immutable 和 append-only 属性。
/var
这个目录不能设置 immutable 属性。对 append-only 属性的使用要根据实际情况。例如，为 var/log 目录下的日志文件设置了 append-only 属 性，会使日志轮换(logrotate)无法进行，但不会造成太大问题，你需要权衡是否使用日志轮换的利弊，以绝对是否对日志文件设置 append-only 属性。再比如，sendmail 程序会定时地截断或者覆盖/var/log/sendmail.st 文件，因此也不能设置 append-only 属性。

在对具有 A 属性的文件进行操作时，A 属性可以提高一定的性能。而 S 属性能够最大限度的保障文件的完整性。

chattr 命令可以通过以下三种方式执行：

chattr +Si test.txt
给 test.txt 文件添加同步和不可变属性。

chattr -ai test.txt
把文件的只扩展(append-only)属性和不可变属性去掉。

chattr =aiA test.txt
使 test.txt 文件只有 a、i 和 A 属性。

最后，每个命令都支持-R 选项，用于递归地对目录和其子目录进行操作。

在 linux 下，有些配置文件是不允许任何人修改的，为了防止被误删除或修改，可以设定该文件的"不可修改位(immutable)"
chattr +i /etc/fstab
如果需要修改文件则：
chattr -i /etc/fstab
以后再修改文件

主机直接暴露在 Internet 或者位于其它危险的环境，有很多 shell 帐户或者提供 HTTP 和 FTP 等网络服务，一般应该在安装配置完成后使用 如下命令：
chattr -R +i /bin /boot /etc /lib /sbin
chattr -R +i /usr/bin /usr/include /usr/lib /usr/sbin

chattr +a /var/log/messages /var/log/secure (...)

文件属性和文件系统属性的关系;
文件系统的特性决定着文件属性的定义和修改，比如我们通过 chattr 来锁定一个文件为不可修改或不可删除时，要用到 chattr 的+i 参数;这在 ext2 和 ext3 文件系统是有效的，但在 reiserfs 文件系统是没有任何效果的;
[root@localhost ~]# chattr +i lsfile.sh
[root@localhost ~]# lsattr lsfile.sh
----i-------- lsfile.sh
[root@localhost ~]# rm -rf lsfile.sh
rm: 无法删除 “lsfile.sh”：不允许的操作

注：如果把 lsfile.sh 变成可修改可删除，应该用-i 参数;

比如在 ext3 或 ext2 文件系统中，我们要让一个文件只能追加内容，但不能删除。应该用 chattr 的+a 参数。

226 ☆stat

查看多个文件的信息
stat file1 file2

227 ☆file

file [选项] 文件名
探测文件类型

file filename

228 ☆cat

使用权限:所有使用者

语法:
cat [-选项] 文件

说明:
把文件串连接后传到基本输出(屏幕或加 > fileName 到另一个文件)
功能 1:在标准输出上显示文件。
该命令功能之一是用来显示文件。它依次读取其后所指文件的内容并将其输出到标准输出。
功能 2:连接两个或多个文件
该命令功能之二是用来将两个或多个文件连接起来。

参数:
-n 或 --number 由 1 开始对所有输出的行数编号
-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号
-s 或 --squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行
-v 或 --show-nonprinting 用一种特殊形式显示控制字符，LFD 与 TAB 除外。
加了-v 选项后，-T 和-E 选项将起作用。其中:
-T 将 TAB 显示为"^I"。该选项需要与-v 选项一起使用。即如果没有使用-v 选项，则这个选项将被忽略。
-E 在每行的末尾显示一个$符。该选项需要与-v 选项一起使用。
-u 输出不经过缓冲区。
-A 等于-vET。
-t 等于-vT。
-e 等于-vE。

$ cat example.txt
则在屏幕上显示出 example.txt 文件的内容。

$ cat -A panda.txt
则在屏幕上显示出 panda.txt 文件的内容，而且如果文件中含有特殊字符的话，一并显示。

$ cat file1 file2 > file3
这样就把文件 filel 和文件 file2 的内容合并起来，放入文件 file3 中。(此时在屏幕上并不能直接看到该命令执行后的结果。若想看到连接后的文件内容，可以再使用"cat file3"。)

需要说明的是，当文件内容过多时，就带来一个问题，因为文本在屏幕上迅速地闪过，用户来不及看清其内容。因此，当文件内容较大时，一般可用 more 等命令分屏显示，以免因屏幕滚动太快而无法看清。

cat -n textfile1 > textfile2
把 textfile1 的文件内容加上行号后输入 textfile2 这个文件里

cat -b textfile1 textfile2 >> textfile3
把 textfile1 和 textfile2 的文件内容加上行号(空白行不加)之后将内容附加到 textfile3

cat
从标准输入输入,标准输出上回显
asfd
asfd
sdfkdl
sdfkdl
sdfsdfksdf
sdfsdfksdf

asdfksdfkd
asdfksdfkd
^d 退出

cat > typedin.txt
This time, when text is typed at the keyboard,
It is not echoed back to the screen.
Instead, it is redirected to the file typedin.txt.
^d
输出到文件

cat > typedin.txt <<EOF

cat >> linuxsir.txt
追加

229☆tac

反转文件中行的位置（上和下）。

230☆more

使用权限:所有使用者

语法:
more [-选项] 文件
more [-dlfpcsu] [-num] [+/pattern] [+linenum] [fileNames..]

说明:
在终端屏幕按屏显示文本文件。该命令一次显示一屏文本，显示满之后，停下来，并在终端底部打印出- - More- - ，系统还将同时显示出已显示文本占全部文本的百分比，若要继续显示，按回车或空格键即可。按 b 键就会往回(back)一页显示，而且还有搜寻字串的功能(与 vi 相似)，使用中的说明文件，请按 h 。

参数:
+num 从第 num 行开始显示
-num 一次显示的行数,定义屏幕大小，为 num 行
-d 提示使用者，在画面下方显示 [Press space to continue, q to quit.] ，如果使用者按错键，则会显示 [Press h for instructions.] 而不是 哔 声
-l 不处理< Ctrl+l >(换页符)。如果没有给出这个选项，则 more 命令在显示了一个包含有< Ctrl+l > 字符的行后将暂停显示，并等待接收命令。即取消遇见特殊字元 ^L(送纸字元)时会暂停的功能

-f 计算行数时，以实际上的行数，而非自动换行过后的行数(有些单行字数太长的会被扩展为两行或两行以上)
-p 显示下一屏之前先清屏。不以卷动的方式显示每一页，而是先清除屏幕后再显示内容
-c 跟 -p 相似，不同的是先显示内容再清除其他旧资料
-s 当遇到有连续两行以上的空白行，就代换为一行的空白行
-u 禁止加下划线,(根据环境变数 TERM 指定的 terminal 而有所不同)
+/pattern 在每个文件显示前搜寻该字串(pattern)，然后从该字串之后开始显示
fileNames 欲显示内容的文件，可为复数个数

执行中的命令

在 more 命令的执行过程中，用户可以使用 more 自己的一系列命令动态地根据需要来选择显示的部分。more 在显示完一屏内容之后，将停下来等待用户输入某个命令。下表列出了 more 指令在执行中用到的一些常用命令，而有关这些命令的完整内容，可以在 more 执行时按 h 查看。这些命令的执行方法是先输入 i(行数)的值，再打所要的命令，不然它会以预设值来执行命令。

Enter 向下 n 行，需要定义，默认为 1 行;
Ctrl+f 向下滚动一屏;
空格键 向下滚动一屏;
Ctrl+b 返回上一屏;
= 输出当前行的行号;
:f 输出文件名和当前行的行号;
v 调用 vi 编辑器;
! 命令 调用 Shell，并执行命令;
q 退出 more

i 空格 若指定 i，显示下面的 i 行;否则，显示下一整屏。
i 回车 若指定 i，显示下面的 i 行;否则，显示下一行。
i^D 按，若指定 i，显示下面的 i 行;否则，往下显示半屏(一般为 11 行)。
id 同 i^D 。
iz 同"i 空格"类似，只是 i 将成为以下每个满屏的缺省行数。
is 跳过下面的 i 行再显示一个整屏。预设值为 1。
if 跳过下面的 i 屏再显示一个整屏。预设值为 1。
i^B 按，往回跳过(即向文件首回跳)i 屏，再显示一个满屏。预设值为 1。
b 与 i^B 相同。
' 回到上次搜索的地方
q 或 Q 退出 more。
＝ 显示当前行号。
v 在当前行启动/usr/bin/vi 对之进行编辑修改。
h 显示各命令的帮助信息。
i/pattern 查找匹配该模式的第 i 行。预设值为 1。
in 查找符合表达式的倒数 i 行。预设值为 1。

! 或 :! 在子 shell 中执行命令。
i:n 在命令行中指定了多个文件名的情况下，可用此命令使之显示第 i 个文件，若 i 过大(出界)，则显示文件名列表中的最后一个文件。
i:p 在命令行中指定了多个文件名的情况下，可用此命令使之显示倒数第 i 个文件。若 i 过大(出界)，则显示第一个文件。
i:f 显示当前文件的文件名和行数。
. 重复上次键人的命令。

显示一个文件的内容，但显示之前先清屏，并且在显示器的最下方显示完整的百分比。
$ more -dc example1.c
执行该命令后，先清屏，然后显示文件 example.c 的内容。

显示一个文件的内容，要求每十行显示一次，且显示之前先清屏。
$ more -c -10 example1.c
执行该命令后，先清屏，然后将以每十行每十行的方式显示文件 example.c 的内容。

more -s testfile 逐页显示 testfile 之文件内容，如有连续两行以上空白行则以一行空白行显示。
more +20 testfile 从第 20 行开始显示 testfile 之文件内容。

[root@localhost ~]# more -dc /etc/profile 注：显示提示，并从终端或控制台顶部显示;
[root@localhost ~]# more +4 /etc/profile 注：从 profile 的第 4 行开始显示;
[root@localhost ~]# more -4 /etc/profile 注：每屏显示 4 行;
[root@localhost ~]# more +/MAIL /etc/profile 注：从 profile 中的第一个 MAIL 单词的前两行开始显示;

231 ☆less

使用权限:所有使用者

语法:
less [选项] 文件名

说明:
less 命令的功能几乎和 more 命令一样，也是用来按页显示文件，不同之处在于 less 命令在显示文件时允许用户既可以向前又可以向后翻阅文件。
同时因为 less 并未在一开始就读入整个文件，因此在遇上大型文件的开启时，会比一般的文书编辑器(如 vi)来的快速。

参数:
-i 搜索时忽略大小写;除非搜索串中包含大写字母;
-I 搜索时忽略大小写，除非搜索串中包含小写字母;
-m 显示读取文件的百分比;

-M 显法读取文件的百分比、行号及总行数;
-n 去掉行号
-N 在每行前输出行号;
-? 本选项显示 less 接收的命令小结。若给出本选项则忽略其他选项，less 保留并在帮助屏后显示
-a 在当前屏幕显示的最后一行后开始查询
-c 从顶行向下全屏重写,从顶部（从上到下）刷新屏幕，并显示文件内容。而不是通过底部滚动完成刷新;
-C 类似-c 但在写之前清屏
-e 第二次到文件尾后自动退出 less。若缺省，唯一退出 less 的方法就是通过 q 命令
-E 第一次到文件尾后自动退出 less
-o 文件：见到本项时，把输入拷贝到文件。这只有在输入文件是一条管道，不是普通文件的情况下应用
-O 文件：类似-o，不要求确认就对已有文件重写
-q 产生相对安静的操作。当试图向文件尾之后或文件头之前滚动时，终端铃不响;在产生其他错误时，如键入非法字符，终端铃响
-Q 产生完全安静的操作,在终端下不响铃;
-s 将多个空行压缩成一个空行
-x n 每次按制表符走 n 格，n 的缺省值是 8。
-p pattern 搜索 pattern;比如在/etc/profile 搜索单词 MAIL，就用 less -p MAIL /etc/profile
-f 强制打开文件，二进制文件显示时，不提示警告;

用 less 命令显示文件时，若需要在文件中往前移动，按 b 键;要移动到用文件的百分比表示的某位置，则指定一个 0 到 100 之间的数，并按 p 即可。

动作
回车键 向下移动一行;
y 向上移动一行;
空格键 向下滚动一屏;
b 向上滚动一屏;
d 向下滚动半屏;
h less 的帮助;
u 向上洋动半屏;
w 可以指定显示哪行开始显示，是从指定数字的下一行显示;比如指定的是 6，那就从第 7 行显示;
g 跳到第一行;
G 跳到最后一行;
p n% 跳到 n%，比如 10%，也就是说比整个文件内容的 10%处开始显示;
/pattern 搜索 pattern ，比如 /MAIL 表示在文件中搜索 MAIL 单词;
v 调用 vi 编辑器;
q 退出 less
!command 调用 SHELL，可以运行命令;比如!ls 显示当前列当前目录下的所有文件;

需要按页显示 test 文件
$ less test

ls -l /usr/bin | less

232 ☆head

如果用户希望查看一个文件究竟保存的是什么内容，可以只查看文件的头几行，而不必浏览整个文件。用 head 命令只显示文件或标准输入的头几行。

语法:
head [-n] 文件

功能:
显示指定文件的前若干行。

说明:该命令显示每个指定文件的前面 n 行。如果没有给出 n 值，缺省设置为 10。
如果没有指定文件，head 就从标准输入读取。

显示文件 example.c 的前 3 行。
$ head -3 example.c

显示多个文件的前几行
head file1 file2

233 ☆tail

tail

同样，如果用户想查看文件的尾部，可以使用 tail 命令。

语法:tail [+/-num] [参数] 文件

说明:
该命令显示一个文件的指定内容。它把指定文件的指定显示范围内的内容显示在标准输出上。如果没有给定文件名，则使用标准输入文件。

参数:
+num 从第 num 行开始显示.
-num 从距文件尾 num 行处开始显示.如果省略 num 参数,系统默认值为 10.

l 以文本行为 num 的计数单位。与参数选项+ num 或- num 选项同时使用时，num 表示要显示的文本行行数。
c 以字节为 num 的计数单位。与参数选项+ num 或- num 选项同时使用时，num 表示要显示的字符数。
(l、c 选项可以省略，系统默认值为 l，即按行计数)。

$ tail -4l example
将显示文件 example 的最后 4 行。

tail -f log

234 ☆od

od(Octal Dump)

用户通常使用 od 命令查看特殊格式的文件内容。通过指定该命令的不同选项可以以十进制、八进制、十六进制和 ASCII 码来显示文件。

语法:
od [选项] 文件…

参数:
-A 指定地址基数，包括:
d 十进制
o 八进制(系统默认值)
x 十六进制
n 不打印位移值
-t 指定数据的显示格式，主要的参数有:
c ASCII 字符或反斜杠序列
d 有符号十进制数
f 浮点数
o 八进制(系统默认值为 02)
u 无符号十进制数
x 十六进制数
除了选项 c 以外的其他选项后面都可以跟一个十进制数 n，指定每个显示值所包含的字节数。

用 ASCII 码和十六进制组合的方式能提供更有价值的信息输出。
$ od - Ax - tcx1 wh5

235 ☆cut

使用权限:所有使用者

语法:
cut -c num1-num2 filename

说明:显示每行从开头算起 num1 到 num2 的文字。

shell>> cat example
test2
this is test1

shell>> cut -c0-6 example ## print 开头算起前 6 个字元
test2
this i

cut -d: -f7 passwd > shells
分隔符是: 打印第 7 列

236 ☆sort

语法:
sort [选项] 文件

说明:
Sort 命令将逐行对文件中的内容进行排序，如果两行的首字符相同，该命令将继续比较这两行的下一字符，如果还相同，将继续进行比较。实际上，sort 命令可以被认为是一个非常强大的数据管理工具，用来管理内容类似数据库记录的文件。
sort 命令对指定文件中所有的行进行排序，并将结果显示在标准输出上。如不指定输入文件或使用"- "，则表示排序内容来自标准输入。

sort 排序是根据从输入行抽取的一个或多个关键字进行比较来完成的。排序关键字定义了用来排序的最小的字符序列。缺省情况下以整行为关键字按 ASCII 字符顺序进行排序。

参数:
-m 若给定文件已排好序，合并文件。
-c 检查给定文件是否已排好序，如果它们没有都排好序，则打印一个出错信息，并以状态值 1 退出。
-u 对排序后认为相同的行只留其中一行。
-o 输出文件 将排序输出写到输出文件中而不是标准输出，如果输出文件是输入文件之一，sort 先将该文件的内容写入一个临时文件，然后再排序和写输出结果。
-d 按字典顺序排序，比较时仅字母、数字、空格和制表符有意义。
-f 将小写字母与大写字母同等对待。
-I 忽略非打印字符。

-M 作为月份比较:"JAN"<"FEB"<...<"DEC"。
-r 按逆序输出排序结果。
+posl-pos2 指定一个或几个字段作为排序关键字，字段位置从 posl 开始，到 pos2 为止(包括 posl，不包括 pos2)。如不指定 pos2，则关键字为从 posl 到行尾。字段和字符的位置从 0 开始。
-b 在每行中寻找排序关键字时忽略前导的空白(空格和制表符)。
-t separator 指定字符 separator 作为字段分隔符。

$ cat text
vegetable soup
fresh vegetables
fresh fruit
lowfat milk

$ sort text
fresh fruit
fresh vegetables
lowfat milk
vegetable soup

用户可以保存排序后的文件内容，或把排序后的文件内容输出至打印机。下例中用户把排序后的文件内容保存到名为 result 的文件中。
$ sort text>result

以第 2 个字段作为排序关键字对文件 example 的内容进行排序。
$ sort -k 1,2 example

对于 file1 和 file2 文件内容反向排序，结果放在 outfile 中，利用第 2 个字段的第一个字符作为排序关键字。
$ sort -r -o outfile -k 1.0,-1.1 example

sort 排序常用于在管道中与其他命令连用，组合完成比较复杂的功能，如利用管道将当前工作目录中的文件送给 sort 进行排序，排序关键字是第 6 个至第 8 个字段。
$ ls -l | sort -k 5,7

sort 命令也可以对标准输入进行操作。例如，如果您想把几个文件文本行合并，并对合并后的文本行进行排序，您可以首先用命令 cat 把多个文件合并，然后用管道操作把合并后的文本行输入给命令 sort，sort 命令将输出这些合并及排序后的文本行。在下面的例子中，文件 veglist 与文件 fruitlist 的文本行经过合并与排序后被保存到文件 clist 中。
$ cat veglist fruitlist | sort > clist

$ sort shells > sorted.shells

sort -nr uniq.sorted.shells

237 ☆uniq

语法:
uniq [选项] 文件

说明:
文件经过处理后在它的输出文件中可能会出现重复的行。例如，使用 cat 命令将两个文件合并后，再使用 sort 命令进行排序，就可能出现重复行。这时可以使用 uniq 命令将这些重复行从输出文件中删除，只留下每条记录的唯一样本。
这个命令读取输入文件，并比较相邻的行。在正常情况下，第二个及以后更多个重复行将被删去，行比较是根据所用字符集的排序序列进行的。该命令加工后的结果写到输出文件中。输入文件和输出文件必须不同。如果输入文件用"- "表示，则从标准输入读取。

参数:
-c 显示输出中，在每行行首加上本行在文件中出现的次数。它可取代- u 和- d 选项。
-d 只显示重复行。
-u 只显示文件中不重复的各行。
-n 前 n 个字段与每个字段前的空白一起被忽略。一个字段是一个非空格、非制表符的字符串，彼此由制表符和空格隔开(字段从 0 开始编号)。
+n 前 n 个字符被忽略，之前的字符被跳过(字符从 0 开始编号)。
-f n 与- n 相同，这里 n 是字段数。
-s n 与+n 相同，这里 n 是字符数。

显示文件 example 中不重复的行。
uniq -u example

显示文件 example 中不重复的行，从第 2 个字段的第 2 个字符开始做比较。
uniq -u - 1 +1 example

uniq -c sorted.shells > uniq.sorted.shells

uniq -c sorted.shells > uniq.sorted.shells

238 ☆split

使用权限:所有使用者

语法:
split [OPTION] [INPUT [PREFIX]]

说明:
将一个文件分割成数个。而从 INPUT 分割输出成固定大小的文件，其档名依序为 PREFIXaa, PREFIXab...;PREFIX 预设值为 `x。若没有 INPUT 档或为 `-，则从标准输入读进资料。

参数:
-b, --bytes=SIZE SIZE[bkm] 值为每一输出文件的大小，单位为 byte。
-C, --line-bytes=SIZE 每一输出档中，单行的最大 byte 数。
-l, --lines=NUMBER NUMBER 值为每一输出档的列数大小。默认值为 1000
-NUMBER 与 -l NUMBER 相同。
--verbose 于每个输出档被开启前，列印出侦错资讯到标准错误输出。
--help 显示辅助资讯然后离开。
--version 列出版本资讯然后离开。
- 从标准输入读取;
SIZE 可加入单位: b 代表 512， k 代表 1K， m 代表 1 Meg。

split -l 2 linuxdoc.txt linuxdocsp 注：切分 linuxdoc.txt 文件，被切分后的文件名 linuxdocsp 开头
ls linuxdocspa* 注：查看切分后的所有文件;

split -b 30 linuxdoc.txt linuxdocwsp 如果指定切分体积大小后面没有接单位，默认是 B;
ls -lh linuxdocwspa*

split -b 3k moretool.txt moretoolwsp 注： 切分后每个文件大小为 3K;

合并文件
cat linuxdocwsp* > newmyfile.img

从标准输入读取
ls -lh /etc | split -l 40 - etcfilelist

PostgresSQL 大型资料库备份与回存:

因 Postgres 允许表格大过你系统文件的最大容量，所以要将表格 dump 到单一的文件可能会有问题，使用 split 进行文件分割。

% pg_dump dbname | split -b 1m - filename.dump.

重新载入
% createdb dbname
% cat filename.dump.* | pgsql dbname

239 ☆tr

tr abcdef...[del] ABCDE...[del]
tr a-z A-Z
tr [:lower:] [:upper:]

去掉不想要的字串
tr -d this

取代字串
tr -s "this" "TEST"

tr 'aeiou' 'AEIOU'
aeaaa
AEAAA

tr 'aeiou' 'AEIOU' > trfile.txt
键盘输入翻译到文件

tr 'aeiou' 'AEIOU' < packages1.txt
文件翻译

tr 'aeiou' 'AEIOU' < packages1.txt >packages1.trfile.txt

tr "A-Z" "a-z" < modified.passwd > modified2.passwd

240 ☆wc

语法:
wc [选项] 文件…

说明:
wc 命令的功能为统计指定文件中的字节数、字数、行数, 并将统计结果显示输出。
如果没有给出文件名，则从标准输入读取。wc 同时也给出所有指定文件的总统计数。
字是由空格字符区分开的最大字符串。

参数:
-c 统计字节数。
-l 统计行数。
-w 统计字数。

输出列的顺序和数目不受选项的顺序和数目的影响。总是按下述顺序显示并且每项最多一列。

行数、字数、字节数、文件名

如果命令行中没有文件名，则输出中不出现文件名。
$ wc - lcw file1 file2
4 33 file1
7 52 file2
11 11 85 total

省略任选项-lcw，wc 命令的执行结果与上面一样。

241 ☆aspell

aspell -l < /usr/share/doc/nautilus-*/NEWS | sort | uniq | wc -l

/usr/share/dict/words
字典文件
/usr/share/dict/words -> linux.words

242 ☆comm

comm

语法:
comm [-123] file1 file2

说明:
如果想对两个有序的文件进行比较，可以使用 comm 命令。
该命令是对两个已经排好序的文件进行比较。其中 file1 和 file2 是已排序的文件。
comm 读取这两个文件，然后生成三列输出:仅在 file1 中出现的行;仅在 file2 中出现的行;在两个文件中都存在的行。如果文件名用"- "，则表示从标准输入读取。

选项 1、2 或 3 抑制相应的列显示。例如 comm - 12 就只显示在两个文件中都存在的行;comm - 23 只显示在第一个文件中出现而未在第二个文件中出现的行;comm - 123 则什么也不显示。

$ comm -12 myfile1 myfile2
就只显示文件 myfile1 和 myfile2 中共有的行。

243 ☆diff

语法:
diff [选项] file1 file2

说明:
该命令的功能为逐行比较两个文本文件，列出其不同之处。它比 comm 命令完成更复杂的检查。它对给出的文件进行系统的检查，并显示出两个文件中所有不同的行，不要求事先对文件进行排序。
该命令告诉用户，为了使两个文件 file1 和 file2 一致，需要修改它们的哪些行。如果用"- "表示 file1 或 file2，则表示标准输入。如果 file1 或 file2 是目录，那么 diff 将使用该目录中的同名文件进行比较。例如:

diff /usr/panda mine
把目录/usr/panda 中名为 mine 的文件与当前目录中的 mine 文件进行比较。

通常输出由下述形式的行组成:
n1 a n3，n4
n1，n2 d n3
n1，n2 c n3，n4

这些行类似 ed 命令把 filel 转换成 file2。字母(a、d 和 c)之前的行号(n1，n2)是针对 file1 的，其后面的行号(n3，n4)是针对 file2 的。
字母 a、d 和 c 分别表示附加、删除和修改操作。

在上述形式的每一行的后面跟随受到影响的若干行，以"＜"打头的行属于第一个文件，以"＞"打头的行属于第二个文件。

diff 能区别块和字符设备文件以及 FIFO(管道文件)，不会把它们与普通文件进行比较。

如果 file1 和 file2 都是目录，则 diff 会产生很多信息。如果一个目录中只有一个文件，则产生一条信息，指出该目录路径名和其中的文件名。

参数:
-b 忽略行尾的空格，而字符串中的一个或多个空格符都视为相等。如 How are you 与 How are you 被视为相同的字符串。
-c 采用上下文输出格式(提供三行上下文)。
-C n 采用上下文输出格式(提供 n 行上下文)。
-e 产生一个合法的 ed 脚本作为输出。
-r 当 file1 和 file2 是目录时，递归作用到各文件和目录上。

$ diff m1.c m2.c

屏幕上显示:
3，5 c 3，6
printf("Hello！ ＼n");

}
＜5
＞3 int n，m;
＞4 n＝10 ;
＞5 printf (" % d ＼n " , m = n * 10);
＞6 }

表示把文件 m1.c 的 3 至 5 行改成 m2.c 的 3 至 6 行后，两个文件相同。

diff modified2.passwd passwd

244 ☆find

语法:
find 起始目录 寻找条件 操作
find pathname -options [-print -exec -ok ...]

使用说明:
在目录结构中搜索文件，并执行指定的操作。此命令提供了相当多的查找条件，功能很强大。将文件系统内符合 expression 的文件列出来。你可以指要文件的名称、类别、时间、大小、权限等不同资讯的组合，只有完全相符的才会被列出来。

find 根据下列规则判断 path 和 expression，在命令列上第一个 - ( ) , ！之前的部份为 path，之后的是 expression。如果 path 是空字串则使用目前路径，如果 expression 是空字串则使用 -print 为预设 expression

说明:find 命令从指定的起始目录开始,递归地搜索其各个子目录,查找满足寻找条件的文件并对之采取相关的操作.

该命令提供的寻找条件可以是一个用逻辑运算符 not,and,or 组成的复合条件.逻辑运算符 and,or,not 的含义为:
(1)and:逻辑与,在命令中用"-a"表示,是系统缺省的选项,表示只有当所给的条件都满足时,寻找条件才算满足.
$ find -name 'tmp' -xtype c -user 'inin'
该命令寻找三个给定条件都满足的所有文件.

(2)or:逻辑或,在命令中用"-o"表示.该运算符表示只要所给的条件中有一个满足时,寻找条件就算满足.例如:

$ find -name 'tmp' -o -name 'mina*'
该命令查询文件名为'tmp'或是匹配'mina*'的所有文件.

(3)not:逻辑非,在命令中用"!"表示.该运算符表示查找不满足所给条件的文件.例如:
$ find ! -name 'tmp'
该命令查询文件名不是'tmp'的所有文件.

需要说明的是:当使用很多的逻辑选项时,可以用括号把这些选项括起来.为了避免 Shell 本身对括号引起误解,在话号前需要加转义字符""来去除括号的意义.
$ find (-name 'tmp' -xtype c -user 'inin')

寻找条件有以下选项:

首先，下列各个选项中的 n 值可以有三种输入方式，假设 n 为 20，则:
+20 表示 20 以后(21，22，23 等)
-20 表示 20 以前(19，18，17 等)
20 表示正好是 20

1.以名称和文件属性查找
-name '字串' 查找文件名匹配所给字串的所有文件,字串内可用通配符*,?,[ ]
-lname '字串' 查找文件名匹配所给字串的所有符号链接文件,字串内可用通配符*,?,[ ]
-iname '字串' 查找文件名匹配所给字串的所有文件,忽略大小写,字串内可用通配符*,?,[ ]
-gid n 查找属于 ID 号为 n 的用户组的所有文件
-group gname 查找属于 gname 的用户组的所有文件
-uid n 查找属于 ID 号为 n 的用户的所有文件
-user uname 查找属于 uname 的用户的所有文件
-group '字串' 查找属于用户组名为所给字串的所有的文件
-nogroup 查找无有效所属组的文件，即该文件所属的组在/etc/groups 中不存在。
-nouser 查找无有效属主的文件，即该文件的属主在/etc/passwd 中不存在。
-empty 查找大小为 0 的目录或文件
-path '字串' 查找路径名匹配所给字串的所有文件,字串内可用通配符*,?,[ ]
-ipath '字串' 查找路径名匹配所给字串的所有文件,字串内可用通配符*,?,[ ],忽略大小写
-depth 在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找。
-prune 使用这一选项可以使 find 命令不在当前指定的目录中查找，如果同时使用-depth 选项，那么-prune 将被 find 命令忽略。
-perm 权限 查找具有指定权限的文件和目录,权限的表示可以如 711,644
-size n[bckw] 查找指定文件大小的文件,n 后面的字符表示单位,缺省为 b,代表 512 字节的块,c 表示字元数，k 表示 kilo bytes，w 是二个位元组。
-type x 查找类型为 x 的文件,x 为下列字符之一:
b 块设备文件
c 字符设备文件
d 目录文件

p 命名管道(FIFO)
f 普通文件
l 符号链接文件(symbolic links)
s socket 文件
-xtype x 与-type 基本相同,但只查找符号链接文件
-fstype 查找位于某一类型文件系统中的文件，这些文件系统类型通常可以在配置文件/etc/fstab 中找到，该配置文件中包含了本系统中有关文件系统的信息。
-mount, -xdev : 不跨越文件系统 mount 点,只检查和指定目录在同一个文件系统下的文件，避免列出其它文件系统中的文件
-pid n : process id 是 n 的文件
-links n： 匹配所有连接数为 n 的文件。
-follow：如果 find 命令遇到符号链接文件，就跟踪至链接所指向的文件。
-cpio：对匹配的文件使用 cpio 命令，将这些文件备份到磁带设备中。

2.以时间为条件查找
-amin n 查找 n 分钟以前被访问过的所有文件
-atime n 查找 n 天以前被访问过的所有文件
-cmin n 查找 n 分钟以前文件状态被修改过的所有文件
-ctime n 查找 n 天以前文件状态被修改过的所有文件
-mmin n 查找 n 分钟以前文件内容被修改过的所有文件
-mtime n 查找 n 天以前文件内容被修改过的所有文件
-anewer file : 比文件 file 更晚被读取过的文件
-cnewer file :比文件 file 更新的文件
-newer file1 ! file2 查找更改时间比文件 file1 新但比文件 file2 旧的文件。
-n 表示文件更改时间距现在 n 天以内，+n 表示文件更改时间距现在 n 天以前

3. 可执行的操作
-exec 命令名称 {} 对符合条件的文件执行所给的 Linux 命令,而不询问用户是否需要执行该命令
{}表示命令的参数即为所找到的文件;命令的末尾必须以" \;"结束,注意{}和\;之间的空格。
-ok 命令名称 {} 对符合条件的文件执行所给的 Linux 命令,与 exec 不同的是,它会询问用户是否需要执行该命令
-ls 详细列出所找到的所有文件
-fprintf 文件名 将找到的文件名写入指定文件
-print 在标准输出设备上显示查找出的文件名
-printf 格式 格式的写法请参考有关 C 语言的书

例 1:查找当前目录中所有以 main 开头的文件,并显示这些文件的内容
$ find . - name 'main*' -exec more {} \;

例 2:删除当前目录下所有一周之内没有被访问过的 a .out 或*.o 文件
$ find . (- name a.out - o - name '*.o')
> - atime +7 -exec rm {} \;

说明如下:

命令中的"."表示当前目录,此时 find 将从当前目录开始,逐个在其子目录中查找满足后面指定条件的文件(和)表示括号(),其中的""称为转义符之所以这样写是由于对 Shell 而言,(和)另有不同的含义,而不是这里的用于组合条件的用途"- name a.out"是指要查找名为 a.out 的文件;"- name '*.o"'是指要查找所有名字以 .o 结尾的文件这两个- name 之间的- o 表示逻辑或(or),即查找名字为 a.out 或名字以 .o 结尾的文件,find 在当前目录及其子目录下找到这佯的文件之后,再进行判断,看其最后访问时间是否在 7 天以前(条件-atime +7),若是,则对该文件执行命令 rm(- exec rm{ };)其中{ }代表当前查到的符合条件的文件名,;则是语法所要求的上述命令中第一行的最后一个是续行符当命令太长而在一行写不下时,可输入一个,之后系统将显示一个>,指示用户继续输入命令
你可以使用 ( ) 将运算式分隔,并使用下列运算
exp1 -and exp2
! expr
-not expr
exp1 -or exp2
exp1, exp2

将目前目录及其子目录下所有延伸档名是 c 的文件列出来
find . -name "*.c"

将目前目录其其下子目录中所有一般文件列出
find . -ftype f

将目前目录及其子目录下所有最近 20 分钟内更新过的文件列出
find . -ctime -20

find . -name "[A-Z]*" -print

find /etc -name "host*" -print

find . -perm 755 -print

find . -perm 006
find . -perm -006 #o=rw,只要权限中有这个就符合,只要这些位符合就可以
find . -perm +006 #o=w 或 o=r,某位符合就可以,相当于"或"操作
-perm mode:文件许可正好符合 mode
-perm +mode:文件许可部分符合 mode
-perm -mode: 文件许可某部分完全符合 mode

find /apps -path "/apps/bin" -prune -o -print

如果希望在/apps 目录下查找文件，但不希望在/apps/bin 目录下查找
使用-prune 选项来指出需要忽略的目录。在使用-prune 选项时要当心，因为如果你同时使用了 -depth 选项，那么-prune 选项就会被 find 命令忽略。
-a 和 -o 都是短路求值，与 shell 的 && 和 || 类似如果 -path "/apps/bin" 为真，则求值 -prune，-prune 返回真，与逻辑表达式为真;否则不求值 -prune，与逻辑表达式为假。如果 -path "/apps/bin" -a -prune 为假，则求值 -print ，-print 返回真，或逻辑表达式为真;否则不求值 -print，或逻辑表达式为真。

避开多个文件夹
find /usr/sam \(-path /usr/sam/dir1 -o -path /usr/sam/file1 \) -prune -o -print
\ 表示引用，即指示 shell 不对后面的字符作特殊解释，而留给 find 命令去解释其意义。

find ~ -user sam -print

find /home -nouser -print

find /apps -group gem -print

find / -nogroup -print

find / -mtime -5 -print

find /var/adm -mtime +3 -print

find -newer httpd1.conf ! -newer temp -ls

find . -newer temp -print

find /etc -type d -print

find . ! -type d -print

find /etc -type l -print

find /home/apache -size 100c -print

find . -size +10 -print
当前目录下查找长度超过 10 块的文件（一块等于 512 字节）

先匹配所有的文件，再在子目录中查找。使用 depth 选项就可以使 find 命令这样做。这样做的一个原因就是，当在使用 find 命令向磁带上备份文件系统时，希望首先备份所有的文件，其次再备份子目录中的文件。

find / -name "CON.FILE" -depth -print

在当前的文件系统中查找文件（不进入其他文件系统），可以使用 find 命令的 mount 选项。
find . -name "*.XC" -mount -print

245 ☆xargs

在使用 find 命令的-exec 选项处理匹配到的文件时，find 命令将所有匹配到的文件一起传递给 exec 执行。但有些系统对能够传递给 exec 的命令长度有限制，这样在 find 命令运行几分钟之后，就会出现溢出错误。错误信息通常是“参数列太长”或“参数列溢出”。这就是 xargs 命令的用处所在，特别是与 find 命令一起使用。

find 命令把匹配到的文件传递给 xargs 命令，而 xargs 命令每次只获取一部分文件而不是全部，不像-exec 选项那样。这样它可以先处理最先获取的一部分文件，然后是下一批，并如此继续下去。

在有些系统中，使用-exec 选项会为处理每一个匹配到的文件而发起一个相应的进程，并非将匹配到的文件全部作为参数一次执行;这样在有些情况下就会出现进程过多，系统性能下降的问题，因而效率不高;

而使用 xargs 命令则只有一个进程。另外，在使用 xargs 命令时，究竟是一次获取所有的参数，还是分批取得参数，以及每一次获取参数的数目都会根据该命令的选项及系统内核中相应的可调参数来确定。

find . -type f -print | xargs file

find . -perm -7 -print | xargs chmod o-w

find . -type f -print | xargs grep "hostname"

246 ☆locate

locate
使用权限:所有使用者

语法:
locate 相关字
locate [-q] [-d] [--database=]
locate [-r] [--regexp=]
locate [-qv] [-o] [--output=]
locate [-e] [-f] <[-l] [-c]
<[-U] [-u]>
locate [-Vh] [--version] [--help]

说明:
locate 命令用于查找文件，它比 find 命令的搜索速度快，它需要一个数据库，这个数据库由每天的例行工作(crontab)程序来建立。当我们建立好这个数据库后，就可以方便地来搜寻所需文件了。

在一般的 distribution 之中，资料库的建立都被放在 contab 中自动执行。一般使用者在使用时只要用
locate your_file_name
的型式就可以了。

参数:
-u 建立资料库，-u 会由根目录开始
-U 建立资料库，-U 则可以指定开始的位置。
-e 将排除在寻找的范围之外。建立库的时候用到
-l 如果 是 1.则启动安全模式。在安全模式下，使用者不会看到权限无法看到的文件。这会始速度减慢，因为 locate 必须至实际的文件系统中取得文件的权限资料。
-f 将特定的文件系统排除在外，例如我们没有到理要把 proc 文件系统中的文件放在资料库中。
-q 安静模式，不会显示任何错误讯息。
-n 至多显示 个输出。
-r 使用正规运算式 做寻找的条件。
-o 指定资料库存的名称。
-d 指定资料库的路径
-h 显示辅助讯息
-v 显示更多的讯息
-V 显示程式的版本讯息

locate chdrv
寻找所有叫 chdrv 的文件

locate -n 100 a.out
寻找所有叫 a.out 的文件，但最多只显示 100 个

locate -u
建立资料库

locate -e /proc -u
排除了/proc

247 ☆slocate

248 ☆updatedb

updatedb 建立索引数据库

249 ☆grep

grep （global search regular expression(RE) and print out the line,全面搜索正则表达式并把行打印出来）是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。Unix 的 grep 家族包括 grep、egrep 和 fgrep。egrep 和 fgrep 的命令只跟 grep 有很小不同。egrep 是 grep 的扩展，支持更多的 re 元字符， fgrep 就是 fixed grep 或 fast grep，它们把所有的字母都看作单词，也就是说，正则表达式中的元字符表示回其自身的字面意义，不再特殊。linux 使用 GNU 版本的 grep。它功能更强，可以通过-G、-E、-F 命令行选项来使用 egrep 和 fgrep 的功能。

grep 的工作方式是这样的，它在一个或多个文件中搜索字符串模板。如果模板包括空格，则必须被引用，模板后的所有字符串被看作文件名。搜索的结果被送到屏幕，不影响原文件内容。

grep 可用于 shell 脚本，因为 grep 通过返回一个状态值来说明搜索的状态，如果模板搜索成功，则返回 0，如果搜索不成功，则返回 1，如果搜索的文件不存在，则返回 2。我们利用这些返回值就可进行一些自动化的文本处理工作。

grep 命令一次只能搜索一个指定的模式;egrep 命令检索扩展的正则表达式（包括表达式组和可选项）;fgrep 命令检索固定字符串，它不识别正则表达式，是快速搜索命令。
grep 命令的搜索功能比 fgrep 强大，因为 grep 命令的搜索模式可以是正则表达式，而 fgrep 却不能。

这组命令在指定的输入文件中查找与模式匹配的行。如果没有指定文件，则从标准输入中读取。正常情况下，每个匹配的行被显示到标准输出。如果要查找的文件是多个，则在每一行输出之前加上文件名。

语法：
grep [选项] [查找模式] [文件名 1，文件名 2，……]
egrep [选项] [查找模式] [文件名 1，文件名 2，……]
fgrep [选项] [查找模式] [文件名 1，文件名 2，……]

参数：
-E 每个模式作为一个扩展的正则表达式对待。
-F 每个模式作为一组固定字符串对待（以新行分隔），而不作为正则表达式。
-b 在输出的每一行前显示包含匹配字符串的行在文件中的字节偏移量。
-c 只显示匹配行的数量。
-i 比较时不区分大小写。
-h 在查找多个文件时，指示 grep 不要将文件名加入到输出之前。

-l 显示首次匹配串所在的文件名并用换行符将其隔开。当在某文件中多次出现匹配串时，不重复显示此文件名。
-n 在输出前加上匹配串所在行的行号（文件首行行号为 1）。
-v 只显示不包含匹配串的行。
-x 只显示整行严格匹配的行。
-e expression 指定检索使用的模式。用于防止以“-”开头的模式被解释为命令选项。
-f expfile 从 expfile 文件中获取要搜索的模式，一个模式占一行。

grep 正则表达式元字符集（基本集）

^
锚定行的开始 如：'^grep'匹配所有以 grep 开头的行。

$
锚定行的结束 如：'grep$'匹配所有以 grep 结尾的行。

.
匹配一个非换行符的字符 如：'gr.p'匹配 gr 后接一个任意字符，然后是 p。

*
匹配零个或多个先前字符 如：'*grep'匹配所有一个或多个空格后紧跟 grep 的行。 .*一起用代表任意字符。

[]
匹配一个指定范围内的字符，如'[Gg]rep'匹配 Grep 和 grep。

[^]
匹配一个不在指定范围内的字符，如：'[^A-FH-Z]rep'匹配不包含 A-R 和 T-Z 的一个字母开头，紧跟 rep 的行。

\(..\)
标记匹配字符，如'\(love\)'，love 被标记为 1。

\<
锚定单词的开始，如:'\<grep'匹配包含以 grep 开头的单词的行。

\>
锚定单词的结束，如'grep\>'匹配包含以 grep 结尾的单词的行。

x\{m\}
重复字符 x，m 次，如：'0\{5\}'匹配包含 5 个 o 的行。

x\{m,\}
重复字符 x,至少 m 次，如：'o\{5,\}'匹配至少有 5 个 o 的行。

x\{m,n\}
重复字符 x，至少 m 次，不多于 n 次，如：'o\{5,10\}'匹配 5--10 个 o 的行。

\w
匹配文字和数字字符，也就是[A-Za-z0-9]，如：'G\w*p'匹配以 G 后跟零个或多个文字或数字字符，然后是 p。

\W
\w 的反置形式，匹配一个或多个非单词字符，如点号句号等。

\b
单词锁定符，如: '\bgrepb\'只匹配 grep。

用于 egrep 和 grep -E 的元字符扩展集

+
匹配一个或多个先前的字符。如：'[a-z]+able'，匹配一个或多个小写字母后跟 able 的串，如 loveable,enable,disable 等。

?
匹配零个或多个先前的字符。如：'gr?p'匹配 gr 后跟一个或没有字符，然后是 p 的行。

a|b|c
匹配 a 或 b 或 c。如：grep|sed 匹配 grep 或 sed

()
分组符号，如：love(able|rs)ov+匹配 loveable 或 lovers，匹配一个或多个 ov。

x{m},x{m,},x{m,n}
作用同 x\{m\},x\{m,\},x\{m,n\}

POSIX 字符类

为了在不同国家的字符编码中保持一至，POSIX(The Portable Operating System Interface)增加了特殊的字符类，如[:alnum:]是 A-Za-z0-9 的另一个写法。要把它们放到[]号内才能成为正则表达式，如[A- Za-z0-9]或[[:alnum:]]。在 linux 下的 grep 除 fgrep 外，都支持 POSIX 的字符类。

[:alnum:]
文字数字字符

[:alpha:]
文字字符

[:digit:]
数字字符

[:graph:]
非空字符（非空格、控制字符）

[:lower:]
小写字符

[:cntrl:]
控制字符

[:print:]
非空字符（包括空格）

[:punct:]
标点符号

[:space:]
所有空白字符（新行，空格，制表符）

[:upper:]
大写字符

[:xdigit:]
十六进制数字（0-9，a-f，A-F）

Grep 命令选项

-?
同时显示匹配行上下的？行，如：grep -2 pattern filename 同时显示匹配行的上下 2 行。

-b，--byte-offset
打印匹配行前面打印该行所在的块号码。

-c,--count
只打印匹配的行数，不显示匹配的内容。

-f File，--file=File
从文件中提取模板。空文件中包含 0 个模板，所以什么都不匹配。

-h，--no-filename
当搜索多个文件时，不显示匹配文件名前缀。

-i，--ignore-case
忽略大小写差别。

-q，--quiet
取消显示，只返回退出状态。0 则表示找到了匹配的行。

-l，--files-with-matches
打印匹配模板的文件清单。

-L，--files-without-match
打印不匹配模板的文件清单。

-n，--line-number
在匹配的行前面打印行号。

-s，--silent
不显示关于不存在或者无法读取文件的错误信息。

-v，--revert-match
反检索，只显示不匹配的行。

-w，--word-regexp
如果被\<和\>引用，就把表达式做为一个单词搜索。

-V，--version
显示软件版本信息。

实例

要用好 grep 这个工具，其实就是要写好正则表达式，所以这里不对 grep 的所有功能进行实例讲解，只列几个例子，讲解一个正则表达式的写法。

$ ls -l | grep '^a'
通过管道过滤 ls -l 输出的内容，只显示以 a 开头的行。

$ grep 'test' d*

显示所有以 d 开头的文件中包含 test 的行。

$ grep 'test' aa bb cc
显示在 aa，bb，cc 文件中匹配 test 的行。

$ grep '[a-z]\{5\}' aa
显示所有包含每个字符串至少有 5 个连续小写字符的字符串的行。

$ grep 'w\(es\)t.*\1' aa
如果 west 被匹配，则 es 就被存储到内存中，并标记为 1，然后搜索任意个字符（.*），这些字符后面紧跟着另外一个 es（\1），找到就显示该行。如果用 egrep 或 grep -E，就不用"\"号进行转义，直接写成'w(es)t.*\1'就可以了。

在命令后键入搜索的模式，再键入要搜索的文件。其中，文件名列表中也可以使用特殊字符，如“*”等，用来生成文件名列表。如果想在搜索的模式中包含有空格的字符串，可以用单引号把要搜索的模式括起来，用来表明搜索的模式是由包含空格的字符串组成。否则，Shell 将把空格认为是命令行参数的定界符，而 grep 命令将把搜索模式中的单词解释为文件名列表中的一部分。

grep 'text file' example

grep data *

使用-f 选项从指定文件中读取要搜索的模式。在文件中，每个搜索模式占一行。如果经常要搜索一组常见字符串时，这个功能非常有用。要搜索的模式放置在文件 mypats 中
$ cat mypats
editor
create
$ grep -f mypats exam

250☆tar

不能用管道,只能用 find 命令.使用 tar 程序打出来的包我们常称为 tar 包，tar 包文件的命令通常都是以.tar 结尾的

tar 可以为文件和目录创建文件。利用 tar，用户可以为某一特定文件创建文件(备份文件)，也可以在文件中改变文件，或者向文件中加入新的文件。tar 最初被用来在磁带上创建文件，现在，用户可以在任何设备上创建文件，如软盘。利用 tar 命令，可以把一大堆的文件和目录全部打包成一个文件，这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。Linux 上的 tar 是 GNU 版本的。

tar 速度比 cpio 慢，且不能跨越两份存储媒体，但文件格式几乎在所有的 Unix 系统中都能通用，且使用简便。

语法:tar [主选项+辅选项] 文件或者目录

使用该命令时，主选项是必须要有的，它告诉 tar 要做什么事情，辅选项是辅助使用的，可以选用。

主选项:
-c 创建新的文件文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。
-r 把要存档的文件追加到文件文件的末尾。例如用户已经作好备份文件，又发现还有一个目录或是一些文件忘记备份了，这时可以使用该选项，将忘记的目录或文件追加到备份文件中。
-t 列出文件文件的内容，查看已经备份了哪些文件。
-u 更新文件。就是说，用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。
-x 从文件文件中释放文件。

辅助选项:
-b 该选项是为磁带机设定的。其后跟一数字，用来说明区块的大小，系统预设值为 20(20*512 bytes)。
-f 使用文件文件或设备，这个选项通常是必选的。
-k 保存已经存在的文件。例如我们把某个文件还原，在还原的过程中，遇到相同的文件，不会进行覆盖。
-m 在还原文件时，把所有文件的修改时间设定为现在。
-M 创建多卷的文件文件，以便在几个磁盘中存放。
-v 详细报告 tar 处理的文件信息。如无此选项，tar 不报告文件信息。
-w 每一步都要求确认。
-z 用 gzip 来压缩/解压缩文件，加上该选项后可以将文件文件进行压缩，但还原时也一定要使用该选项进行解压缩。
-j 用 bzip2 来压缩/解压缩文件，加上该选项后可以将文件文件进行压缩，但还原时也一定要使用该选项进行解压缩。
-Z 用 compress 来压缩/解压缩文件，加上该选项后可以将文件文件进行压缩，但还原时也一定要使用该选项进行解压缩。

要将文件备份到一个特定的设备，只需把设备名作为备份文件名。

用户在/dev/fd0 设备的软盘中创建一个备份文件，并将/home 目录中所有的文件都拷贝到备份文件中。
$ tar cf /dev/fd0 /home

要恢复设备磁盘中的文件，可使用 xf 选项:
$ tar xf /dev/fd0

如果用户备份的文件大小超过设备可用的存贮空间，如软盘，您可以创建一个多卷的 tar 备份文件。M 选项指示 tar 命令提示您使用一个新的存贮设备，当使用 M 选项向一个软驱进行存档时，tar 命令在一张软盘已满的时候会提醒您再放入一张新的软盘。这样您就可以把 tar 文件存入几张磁盘中。
$ tar cMf /dev/fd0 /home

要恢复几张盘中的文件，只要将第一张放入软驱，然后输入有 x 和 M 选项的 tar 命令。在必要时您会被提醒放入另外一张软盘。
$ tar xMf /dev/fd0

tar -cvf all.tar *.jpg
这条命令是将所有.jpg 的文件打成一个名为 all.tar 的包。
-c 是表示产生新的包，-f 指定包的文件名。

tar -rvf all.tar *.gif
这条命令是将所有.gif 的文件增加到 all.tar 的包里面去。
-r 是表示增加文件的意思。

tar -uvf all.tar logo.gif
这条命令是更新原来 tar 包 all.tar 中 logo.gif 文件，
-u 是表示更新文件的意思。

tar -tvf all.tar
这条命令是列出 all.tar 包中所有文件，
-t 是列出文件的意思

tar -xvf all.tar
这条命令是解出 all.tar 包中所有文件，对于.tar 结尾的文件
-x 是解开的意思

tar 调用 gzip
gzip 是 GNU 组织开发的一个压缩程序，.tgz 或.tar.gz 结尾的文件就是 gzip 压缩的结果。与 gzip 相对的解压程序是 gunzip。tar 中使用-z 这个参数来调用 gzip。
tar -czvf all.tar.gz *.jpg
这条命令是将所有.jpg 的文件打成一个 tar 包，并且将其用 gzip 压缩，生成一个 gzip 压缩过的包，包名为 all.tar.gz

tar -xzvf all.tar.gz
这条命令是将上面产生的包解开。

tar 调用 bzip2

bzip2 是一个压缩能力更强的压缩程序，.bz2 结尾的文件就是 bzip2 压缩的结果。与 bzip2 相对的解压程序是 bunzip2。tar 中使用-j 这个参数来调用 bzip2。下面来举例说明一下:
tar -cjvf all.tar.bz2 *.jpg
这条命令是将所有.jpg 的文件打成一个 tar 包，并且将其用 bzip2 压缩，生成一个 bzip2 压缩过的包，包名为 all.tar.bz2
tar -xjvf all.tar.bz2
这条命令是将上面产生的包解开。

tar 调用 compress
compress 也是一个压缩程序，但是好象使用 compress 的人不如 gzip 和 bzip2 的人多。.Z 结尾的文件就是 bzip2 压缩的结果。与 compress 相对的解压程序是 uncompress。tar 中使用-Z 这个参数来调用 gzip。下面来举例说明一下:
tar -cZvf all.tar.Z *.jpg
这条命令是将所有.jpg 的文件打成一个 tar 包，并且将其用 compress 压缩，生成一个 uncompress 压缩过的包，包名为 all.tar.Z
tar -xZvf all.tar.Z
这条命令是将上面产生的包解开

251☆cpio

cpio 则由于可通过管道功能，使得其打包时的文件选择、排除功能非常强，且能跨越多份媒体，并能备份特殊的系统文件。

A)含子目录打包:
find /usr/lib -print|cpio -o>/u0/temp1.cpio
将/usr /lib 目录下的文件与子目录打包成一个文件库为/u0/temp1.cpio

若通过-o 选项来打包整个目录下的所有文件与子目录,常先利用 find 目录名-print 来找出所有文件与子目录的名称,通过管道"|"传给 cpio 打包.

B)不含子目录的打包:
ls /usr/lib/*|cpio -o>/u0/temp1.cpio
将/usr/lib 目录下的文件(不含子目录下的文件)打包成一个文件库为/u0/temp1.cpio

C)特定文件打包:
可利用文本搜索命令 grep 与管道配合,可以排除或选择特定类型的文件传给 cpio 打包.
如:ls /usr/lib/*.c|cpio -o>/u0/temp1.cpio
或 ls /usr/lib|grep ′ \.c$′ |cpio -o>/u0/temp1.cpio
意思均为找出/usr/lib 目录下以.c 结尾的文件予以打包.

又如:ls /usr/lib|grep abcd|cpio -o>/u0/temp1.cpio
其意为找出/usr/lib 目录下文件名中含有 abcd 字符的文件予以打包.

ls /usr/lib|grep -v abcd|cpio -o>/u0/temp1.cpio,其意为找出/usr/lib 目录下文件名中不含 abcd 字符的文件予以打包.

-v 选项在 grep 命令中的意思是排除含有字符串的行列.

解包
若以相对路径打包的,当解包展开时,也是以相对路径存放展开的文件数据;若以绝对路径打包的,当解包展开时,也是以绝对路径存放展开的文件数据.因此注意若为相对路径,应先进入相应的目录下再展开.

cpio -id</u0/tmp1.cpio
则将/u0/temp1.cpio 解压到/u1 下(这里假设 temp1.cpio 以相对路径压缩).

若加 u 选项,如 cpio -iud</u0/temp1.cpio 则文件若存在将被覆盖,即强制覆盖.

cpio -id *.c </u0/temp1.cpio
则展开其中的*.c 文件

显示:
cpio -it *.c </u0/temp1.cpio
显示文件库内的文件名称

cpio 命令有三种模式:
copy-out 将系统的文件生成一个 cpio 格式的包
copy-in 将一个 cpio 格式的包解开
copy-pass 相当于一个 cp 命令，不进行任何的压包和解包的工作，经常用来复制文件分别对应的参数是-o,-i,，-p 这样记忆就比较方便了,linux 命令的参数虽然很多，但是还是有规律可以寻找的。

创建:
原始形态:
[root@laptop tmp]# cpio -o -O photo.cpio
按回车以后要手工输入你所需要打包的文件名。以 Ctrl+D 组合键结束。

[root@laptop tmp]# ls
Dcp_0803.jpg Dcp_0810.jpg Dcp_0815.jpg Dcp_0820.jpg Dcp_0827.jpg
Dcp_0804.jpg Dcp_0811.jpg Dcp_0816.jpg Dcp_0821.jpg Dcp_0828.jpg
Dcp_0805.jpg Dcp_0812.jpg Dcp_0817.jpg Dcp_0822.jpg
Dcp_0808.jpg Dcp_0813.jpg Dcp_0818.jpg Dcp_0825.jpg
Dcp_0809.jpg Dcp_0814.jpg Dcp_0819.jpg Dcp_0826.jpg

[root@laptop tmp]# ls | cpio -o > photo.cpio

cpio: Dcp_0803.jpg: truncating inode number
cpio: Dcp_0804.jpg: truncating inode number
cpio: Dcp_0805.jpg: truncating inode number
cpio: Dcp_0808.jpg: truncating inode number
cpio: Dcp_0809.jpg: truncating inode number
cpio: Dcp_0810.jpg: truncating inode number
cpio: Dcp_0811.jpg: truncating inode number
cpio: Dcp_0812.jpg: truncating inode number
cpio: Dcp_0813.jpg: truncating inode number
cpio: Dcp_0814.jpg: truncating inode number
cpio: Dcp_0815.jpg: truncating inode number
cpio: Dcp_0816.jpg: truncating inode number
cpio: Dcp_0817.jpg: truncating inode number
cpio: Dcp_0818.jpg: truncating inode number
cpio: Dcp_0819.jpg: truncating inode number
cpio: Dcp_0820.jpg: truncating inode number
cpio: Dcp_0821.jpg: truncating inode number
cpio: Dcp_0822.jpg: truncating inode number
cpio: Dcp_0825.jpg: truncating inode number
cpio: Dcp_0826.jpg: truncating inode number
cpio: Dcp_0827.jpg: truncating inode number
cpio: Dcp_0828.jpg: truncating inode number
2792 blocks

注：-o 参数也可义用 --create 替代，表示正在使用 copy-out 模式，
-O 参数只能和-o 一起使用，表示出于 out 模式时，指定生成的文件。

root@laptop tmp]# ls | cpio -o -O photo.cpio
这个命令，是等效的，只不过前者用重定向符号实现的而已。由此可以体现 Linux 命令的复杂还有变化多样，所以要多思考。

查看，解包：

解包

[root@laptop tmp]# cpio -i -I photo.cpio
2792 blocks

注：-i 参数是表示使用 copy-in 模式
查看一个 cpio 包里的文件列表的参数和 tar 的参数有点相像，都是用-t 参数，
也可以用--list 替代。
-I 参数只和-t , -i 两个参数连用，表示在 copy-in 模式时指定一个 cpio 包。

有点像 tar 包里的-f 参数，无论是打 tar 包还是解 tar 包或者是查看 tar 的内容,都要加-f 参数一样的道理,在这里，cpio 只不过分成了 copy-in 和 copy-out 两个部分而已。

```
[root@laptop tmp]# cpio -t -I photo.cpio
Dcp_0803.jpg
Dcp_0804.jpg
Dcp_0805.jpg
Dcp_0808.jpg
Dcp_0809.jpg
Dcp_0810.jpg
Dcp_0811.jpg
Dcp_0812.jpg
Dcp_0813.jpg
Dcp_0814.jpg
Dcp_0815.jpg
Dcp_0816.jpg
Dcp_0817.jpg
Dcp_0818.jpg
Dcp_0819.jpg
Dcp_0820.jpg
Dcp_0821.jpg
Dcp_0822.jpg
Dcp_0825.jpg
Dcp_0826.jpg
Dcp_0827.jpg
Dcp_0828.jpg
2792 blocks
```

查看一个 cpio 包里的文件列表的参数和 tar 的参数有点相像，都是用-t 参数，这里可以用--list 替代。

还有一些高级的应用参数：

-u

解包时如果文件已经在也强行覆盖

```
[root@laptop tmp]# cpio -i -I test.cpio -u
[root@laptop tmp]# cpio -i -u -I test.cpio
```

-f

文件匹配参数，但是是一个反向的，就是符合这个指定的文件不解包

```
[root@laptop tmp]# cpio -i -I test.cpio -f *.rpm
```

这条命令的结果是 test.cpio 里除了*.rpm 文件都会解包出来。

-L
这个参数是在打包的时候用的，如果被打包的文件中有符号连接的话，这个参数将会把连接的目标文件打入包中，而不是仅仅把符号连接打包。
[root@laptop tmp]# ls | cpio -o -O newtest.cpio -L

-R
这个参数只用在解包的时候，是用来给解包出来的文件设置所有者和所属组的。但是用户和组的关系必须已经在 passwd,group 两个文件中存在！！！
[root@laptop tmp]# cpio -i -I test.cpio -R yue.pye
[root@laptop tmp]# cpio -i -I test.cpio -R yue:pye
这里是两种表示用户组之间关系的方式一个是.还有一个是:

-r
这个参数只在解包是有用，而且不能和-u 参数同时使用，否则失效，他的作用是当包中的文件和解包所在目录下的某个文件重名的时候自动提示是否需要更名。
[root@laptop tmp]# cpio -i -I test.cpio -r
rename 12.1 -> a
rename backgroud -> c
rename bin -> bindir
rename crystal ->

-A
这个参数只用在压包的时候，用来向已经存在的 cpio 包添加文件的
[root@laptop tmp]# cpio -o -O test.cpio -A
[root@laptop tmp]# cpio -o -A -O test.cpio

252 ☆dump

是一个专门用来备份的工具。
功能说明：备份文件系统。

语法：dump [-cnu][-0123456789][-b <区块大小>][-B <区块数目>][-d <密度>][-f <设备名称>][-h <层级>][-s <磁带长度>][-T <日期>][目录或文件系统] 或 dump [-wW]

补充说明：dump 为备份工具程序，可将目录或整个文件系统备份至指定的设备，或备份成一个大文件。
参数：
-0123456789 备份的层级。
-b<区块大小> 指定区块的大小，单位为 KB。
-B<区块数目> 指定备份卷册的区块数目。
-c 修改备份磁带预设的密度与容量。

-d<密度> 设置磁带的密度。单位为 BPI。
-f<设备名称> 指定备份设备。
-h<层级> 当备份层级等于或大雨指定的层级时，将不备份用户标示为"nodump"的文件。
-n 当备份工作需要管理员介入时，向所有"operator"群组中的使用者发出通知。
-s<磁带长度> 备份磁带的长度，单位为英尺。
-T<日期> 指定开始备份的时间与日期。
-u 备份完毕后，在/etc/dumpdates 中记录备份的文件系统，层级，日期与时间等。
-w 与-W 类似，但仅显示需要备份的文件。
-W 显示需要备份的文件及其最后一次备份的层级，时间与日期。

dump 文件|磁盘 需要备份的目标

备份到/dev/hda3 磁盘中
[root@laptop pyegrp]# dump -0 -u -f /dev/hda3 /home/pyegrp/
DUMP: Date of this level 0 dump: Sat Nov 30 19:45:56 2002
DUMP: Dumping /dev/hda9 (/home/pyegrp) to /dev/hda3
DUMP: Added inode 7 to exclude list (resize inode)
DUMP: Label: none
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 15492 tape blocks.
DUMP: Volume 1 started with block 1 at: Sat Nov 30 19:46:01 2002
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /dev/hda3
DUMP: Volume 1 completed at: Sat Nov 30 19:46:13 2002
DUMP: Volume 1 15490 tape blocks (15.13MB)
DUMP: Volume 1 took 0:00:12
DUMP: Volume 1 transfer rate: 1290 kB/s
DUMP: 15490 tape blocks (15.13MB) on 1 volume(s)
DUMP: finished in 12 seconds, throughput 1290 kBytes/sec
DUMP: Date of this level 0 dump: Sat Nov 30 19:45:56 2002
DUMP: Date this dump completed: Sat Nov 30 19:46:13 2002
DUMP: Average transfer rate: 1290 kB/s
DUMP: DUMP IS DONE

-0 是备份的等级，从 0-9，0 以后都是更新备份的参数，一般常用的是 0 和 9
-u 是将备份记录写入/etc/dumpdates 这个文件中,必须是完整的文件系统

[root@laptop pyegrp]# cat /etc/dumpdates
/dev/hda9 0 Sat Nov 30 19:45:56 2002
/dev/hda9 9 Sat Nov 30 17:04:20 2002

注意，当你备份的目录是一个子目录，而不是一个分区的顶层目录时-u 参数是不能使用的。

-f 是指定备份到那个文件,也可以指定一个设备文件，我在这里指定的就是一个分区

更新备份：
在 0 级别备份的基础上，向/home/pyegrp 写入一些新的文件，然后
umount /home/pyegrp
mount /home/pyegrp
注意如果不这样做，不会将新写入的文件备份进去的。
然后：
[root@laptop pyegrp]# dump -9 -u -f /dev/hda3 /home/pyegrp/
用 restore tf /dev/hda3
就会看到刚刚写入的文件会出现在列表中。

253 ☆restore

是用来恢复备份的。
功能说明：还原(Restore)由倾倒(Dump)操作所备份下来的文件或整个文件系统(一个分区)。

语法：restore [-cCvy][-b <区块大小>][-D <文件系统>][-f <备份文件>][-s <文件编号>] 或 restore [-chimvy][-b <区块大小>][-f <备份文件>][-s <文件编号>] 或 restore [-crvy][-b <区块大小>][-f <备份文件>][-s <文件编号>] 或 restore [-cRvy][-b <区块大小>][-D <文件系统>][-f <备份文件>][-s <文件编号>] 或 restore [chtvy][-b <区块大小>][-D <文件系统>][-f <备份文件>][-s <文件编号>][文件...] 或 restore [-chmvxy][-b <区块大小>][-D <文件系统>][-f <备份文件>][-s <文件编号>][文件...]

补充说明：restore 指令所进行的操作和 dump 指令相反，倾倒操作可用来备份文件，而还原操作则是写回这些已备份的文件。

参数：
-b<区块大小> 设置区块大小，单位是 Byte。
-c 不检查倾倒操作的备份格式，仅准许读取使用旧格式的备份文件。
-C 使用对比模式，将备份的文件与现行的文件相互对比。
-D<文件系统> 允许用户指定文件系统的名称。
-f<备份文件> 从指定的文件中读取备份数据，进行还原操作。
-h 仅解出目录而不包括与该目录相关的所有文件。
-i 使用互动模式，在进行还原操作时，restore 指令将依序询问用户。
-m 解开符合指定的 inode 编号的文件或目录而非采用文件名称指定。
-r 进行还原操作。
-R 全面还原文件系统时，检查应从何处开始进行。
-s<文件编号> 当备份数据超过一卷磁带时，您可以指定备份文件的编号。
-t 指定文件名称，若该文件已存在备份文件中，则列出它们的名称。

-v 显示指令执行过程。
-x 设置文件名称，且从指定的存储媒体里读入它们，若该文件已存在在备份文件中，则将其还原到文件系统内。
-y 不询问任何问题，一律以同意回答并继续执行指令。

[root@laptop pyegrp]# restore rf /dev/hda3 /home/pyegrp

使用互交模式
restore if /dev/hda3

用 restore 命令来查看备份文件里的文件列表
[root@laptop pyegrp]# restore -tf /dev/hda3

提取到当前目录
restore xf /dev/hda3

254 ☆dump/restore 恢复单个文件

1. 准备用 dump 备份/boot 目录下的文件. 使用 df /boot 查看/boot 所在的设备(以下假设为/dev/hda1)
2. 首先确认备份需要的空间. 查看一个 0 级备份需要的字节数,使用 -S
dump -oS /dev/hda1
3. 备份到文件而非磁带. 确认在/var/tmp 目录是否有足够的空间,执行
dump -0u -f /var/tmp/dumpfile /dev/hda1
4. 检查/etc/dumpdates,查看完全备份的时间戳.
5. 使用 restore 检查备份文件的内容
restore -tf /var/tmp/dumpfile
6. 我们可以使用 restore 的互动模式恢复特定文件到一个临时目录.
mkdir /tmp/restored; cd /tmp/restored
restore -if /var/tmp/dumpfile
7. 这时会看到一个 restore > 提示符. 键入 help 查看可用命令的列表. 使用 ls 和 cd 命令查看备份文件的列表.
8. 使用 add,选中/grub.menu.1st 和/grub/grub.conf 文件.列出所在目录,恢复的文件应该带有星号.
9. 键入 extract 命令恢复选中的文件.设置下个卷名为 1, 不为解压目录设置所有者模式. quit 退出 restore 模式.
10. 在 restore 运行的目录中应该有一个 grub 目录,包含恢复的 grub.conf 和 menu.1st 文件.

255 ☆备份恢复实例

不要忘记 mtime,atime,以及 ctime.
unix 系统为每个文件都记录这三个不同的时间，第一个是 mtime,即修改时间。无论何时，只要文件内容被改变，mtime 的值就会被相应修改。第二个是 atime，即访问时间。只要文件被访问（比

如运行或读取），它就会被修改。第三个是 ctime，即变更时间。当文件的属性发生变化（比如改变权限或者所有关系）时，ctime 的值就会被改变。管理员用 ctime 来查找黑客。
备份会改变 atime，tar,cpio,dd 都会这样做，dump 通过原始设备来读取文件系统，因此它不会改变 atime.

dump 的语法。
dump levelunbdsf blkg-factor density size device-name file_system
例子：
把/home 完全备份到一个称做/dev/rmt/0cbn 的本地磁带驱动器上。
dump 0unbdsf 126 141000 11500 /dev/rmt/0cbn /home
把/home 完全备份到一个称做/backup/home.dump 的光学 CD 驱动器上。
dump 0unbdsf 126 141000 11500 /backup/home.dump /home
以上命令由三个不需要参数的选项（0，u and n）以及四个需要一个协作参数的选项（b,d,s and f）组成。
命令选项：
0---9　　指定 DUMP 应该进行的备份级别。
b　　指定 DUMP 就应该使用的块因子。
u　　指定 DUMP 更新 dumpdates 文件。
n　　完成操作时要通知操作员组的成员。
d（密度） and s（大小）　　告诉 DUMP 备份卷有多大，DUMP 用这些数字来估计要使用何种磁带。
f　　告诉 DUMP 要使用什么设备。
W,w　　告诉 DUMP 执行一次空运行来告诉你什么文件系统需要备份。

要避免跨卷 DUMP。

restore 的语法
restore [trxi]vbsfy blocking-factor file-number device-name
例子：
要恢复一个使用块因子 32 创建的 DUMP 磁带备份（位于/dev/rmt/0cbn）的全部内容。
restore rvbfy 32 /dev/rmt/0cbn
有一个使用块因子 32 创建的位于/dev/rmt/0cbn 中的 DUMP 磁带，如果想从该磁带中恢复文件/etc/hosts and /etc/passwd.
restore xvbfy 32 /dev/rmt/0cbn ./etc/hosts ./etc/passwd
创建内容表
restore tfy device >/tmp/dump.list
命令选项：
决定 restore 的类型
t　　显示卷的内容。
r　　指明卷的整个内容应该被恢复到当前工作目录下。
x　只提取命令后面所列的文件。
i　允许执行交互式恢复。

决定 restore 的行为
v　指定详细输出
s　读取之前跳过的磁带文件个数
b　块因子
f　备份驱动器的文件名。
y　恢复过程禁止询问。

使用 cpio 工具进行备份和恢复
备份的语法
cpio -o[aBcv]
恢复的语法
cpio -i[Btv][patterns]
例子：
在一个本地磁带上创建/home 的一个完全备份。
cd /home
touch level.0.cpio.timestamp　　使增量备份成为可能。建立时间参考点。
find . -print|cpio -oacvB > device　　device 可以是一个光学或 CD 设备。
touch level.1.cpio.timestamp　　增量备份的时间参考点。
find . newer level.1.cpio.timestamp -print|cpio -oacvB >device
在远程磁带上创建/home 的一个完全备份。
cd /home
find . -print|cpio -oavB|(rsh remote_system dd of=device bs=5120)

cpio 备份用绝对路径（find　/home/file）会限制恢复时的灵活性。它只能恢复到/home/file。如果用相对路径（find home/file）就能把它恢复到任何地方。
命令选项
备份类
o　创建一个备份
a　把 atime 重置成备份前的值
c　用 ASCII 首部格式
v　用于详细信息输出
B，C　　指定块大小，它们是互斥的。
恢复类
i　指定输入模式，必须是命令列表的第一个。
t　生成内容表，并不实际操作。
k　跳过坏区
d　在需要时创建目录
m　恢复文件备份时的原始修改时间，否则的话，默认动作是把恢复后的文件的修改时间设为新修改时间
u　无条件覆盖所有文件

"*pattern"　恢复匹配该模式的文件
f "*pattern"　恢复不匹配该模式的文件
r　交互式重命名文件名

2004/02/12
tar 命令的基本语法
tar -[cx]vf device pattern

命令选项
c　创建一个存档
v　详细信息输出
W　对存档文件进行校验
b　块因子
f　输出到 DEVICE 参数所指的设备，DEVICE 可以是文件，光盘，磁带或者标准输出（stdout）
pattern 模式匹配如“a*”
x　恢复一个存档
m　正常情况下，恢复后的文件会保留它们在存档前的修改时间，选用该选项把修改时间改成恢复时间，这与 CPIO 命令的行为完全相反。
o　把恢复的文件的所有者设置成你。这是对于非 root 用户的默认行为，除非使用该选项，否则，root 提取的文件都会归保存在 tar 文档中的用户和组所有。
p　默认情况下，tar 不会恢复所有的文件属性。文件的许可是由当前 umask 决定，而不是由原始文件的许可决定，包括 setuid and sticky 位。这个选项告诉 tar 用原始文件的许可。
d　能够对存档和文件系统进行一个 diff 比较。
a　重设置访问时间
F　在卷结束时运行一个脚本，这可用于进行自动卷交换。
Z z　自动调用 compress and gzip 程序

dd 命令基本语法
dd if=device of=device bs=blocksize
if=　指定输入文件，即 dd 从中拷贝数据的文件。它可以是需要备份的文件或者原始分区，如果从 stdin 中读取数据，那么该参数不必指定。
of=　指定输出文件，即 dd 发送数据的目的地。它可以是需要备份的文件或者原始分区，如果从 stdout 中读取数据，那么该参数不必指定。
bs　指定块大小，即一次 i/o 操作中传输的数据量。

使用 dd 和 rsh and ssh 进行远程备份(GNU tar and GNU cpio 命令能读取远程设备)
读取远程设备上的备份
rsh remote_host "dd if=device ibs=blocksize" |tar xvBf -
ssh remote_host "dd if=device bs=blocksize" |tar xvBf -
ssh remote_host "dd if=device bs=blocksize" |restore rvf -
ssh remote_host "dd if=device bs=blocksize" |cpio -itv

把备份写到远程设备上
tar -cvf - . |(rsh remote_system dd of=device obs=block_size)
dump 0bdsf 64 100000 100000 - |ssh remote_host "dd if=device bs=64k"
tar -cvf - |ssh remote_host "dd if=device bs=10k"
cpio -oacvB |ssh remote_host "dd if=device bs=5k"

LINUX 裸机恢复方法：
1、备份重要的元数据　　# fdisk -l >/etc/fdisk-l.txt
2、用本地工具备份系统　# cd / ; tar cf - . |gzip -c >/backup/xxx.tar.gz
3、系统损坏，用其它介质引导系统（引导盘，KNOPPIX 等）。
4、用元数据对硬盘这行分区并格式化。
解出元数据文件 #gzip -dc /xxx/xxx.tar.gz|tar -xvf - ./etc/fstab ./etc/fdisk-l.txt
分区　# fdisk /dev/sda
创建文件系统 # mke2fs /dev/sda1
5、恢复操作系统信息 # gzip -dc /xxx/xxx.tar.gz|tar xf -
6、在新根磁盘上恢复引导块

ORACLE（离线）冷备份
1、关闭数据库，中止所有允许访问数据库的进程。
2、通过备份工具对文件进行备份（TAR,DD,CPIO）

ORACLE（在线）热备必要步骤
1、请求 ORACLE 的所有表空间及数据文件的列表。
2、请求存储 ORACLE 归档日志的位置。
3、请求存储 ORACLE 控制文件的位置（可选）。
4、将所有表空间置于备份模式，可且用 ALTER TABLESPACE tablespace_name BEGIN BACKUP 命令。
5、将每个表空间的数据文件复制到磁盘或磁带上。
6、去除各个表空间的备份模式，可使用 ALTER TABLESPACE tablespace_name END BACKUP 命令。
7、切换重做日志文件。
8、备份控制文件，可使用 BACKUP CONTROL file 命令。
9、手工复制控制文件（可选）。
10、手工复制在线重做日志。
11、确保所有在备份期间的归档重做日志均保存完好。

说明：

在表空间被置于备份模式时，会有以下事件发生：
1、ORACLE 检查点表空间，将所有改变从内存存储到磁盘上。
2、表空间中各个数据文件的 SCN 标识都“冻结”在当前值，即使对数据文件进一步更新，SCN 值都不会被更新，直到去除备份模式。
3、ORACLE 从记录完整的更改数据库块的映像转向记录重做日志。不再记录某特定的块是怎样改变的，而是记录整个改变后块的映像。这就是重做日志在热备份过程中迅速增长的原因。

自动备份的 ORABACK.SH 脚本支持特性：
1、备份磁盘或磁带。
2、自动检测数据库配置。
3、基于文件系统或原始分区备份数据库。
4、多任务，可以将备份时间减少到 75%。
5、使用邮件发送成功或错误通知。
6、备份 ORATAB 中的一个或多个实例。

使用方法介绍
1、备份所有实例;
ORABACK.SH
2、备份一个或更多实例，要带上参数 ORACLE_SID;
ORABACK.SH ORACLE_SID1 ORACLE_SIDn
3、如果要在 ORABACK.CONF 中指定的某一时刻调度备份，则带上 at:
ORABACK.SH at ORACLE_SID1 ORACLE_SIDn
安装 ORABACK.SH,首先将 ORABACK.SH、CONFIG.GUESS 以及 LOCALPATH.SH 放在一个目录下，然后核对脚本头部特定位置的下列值。
BINDIR　　安装 ORABACK.SH 的目录。
ORATAB　　设为 ORACLE 的 ORATAB 文件的名称和位置。
ORACONF　　设为 ORABACK.CONF 文件的名称和位置。

ORABACK.CONF 配置
1、HOSTNAME.MASTER　系统主机名，去掉域名（如 AAA.DOMAIN.COM 变成 AAA）
2、SKIP　　如果今天晚上跳过该主机上的所有备份，可以在这里输入“SKIP”
3、COLD DAY　　进行冷备份的日期，可以是每周的某一天（FRI,即星期五），或每月的某一天（03，即第三天）。
4、COLD TIME　　进行冷备份的时间，采用 24 小时制。
5、HOT TIME　　一天中进行热备份的时间，采用 24 小时制。
6、TYPE DEVICE　　非回绕磁带设备，备份到磁带。（如只允许磁盘备份，则保持空白）。
7、USERS　　允许运行脚本的用户名列表，以|隔开，如 ORACLE|DBA,空白表示仅允许 ORACLE 用户运行。
8、PARALLELISM　　同时运行数据文件拷贝数，空缺 = 1.
9、BACKUP DIR　　备份目录。
10、Y　　表示在写入磁盘前先对文件进行压缩。

11、MAIL DS　　邮件 ID 列表，用来发关备份成功与否的通知，之间用“，”号分开。

进行完全逻辑备份时需要 RESTRICT 模式。关闭数据库，然后用 STARTUP RESTRICT OPEN 打开数据库，再进行完全导出。导出完成后用 ALTER DATABASE DISABLE RESTRICTED SESSION 恢复连接。

集萃

镜像重做日志　一个 ACTIVE 或 CURRENT 日志组的所有成员均丢失，会造成数据丢失。
镜像重做日志，所有日志组成员均丢失的可能性就极小。
观察告警日志　镜像控制文件　使用 ARCHIVELOG 模式

256 ☆gzip

减少文件大小有两个明显的好处，一是可以减少存储空间，二是通过网络传输文件时，可以减少传输的时间。gzip 是在 Linux 系统中经常使用的一个对文件进行压缩和解压缩的命令，既方便又好用。

语法:gzip [选项] 压缩(解压缩)的文件名

各选项的含义:

-c 将输出写到标准输出上，并保留原有文件。
-d 将压缩文件解压。
-l 对每个压缩文件，显示下列字段:
压缩文件的大小
未压缩文件的大小
压缩比
未压缩文件的名字
-r 递归式地查找指定目录并压缩其中的所有文件或者是解压缩。
-t 测试，检查压缩文件是否完整。
-v 对每一个压缩和解压的文件，显示文件名和压缩比。
-num 用指定的数字 num 调整压缩的速度，-1 或--fast 表示最快压缩方法(低压缩比)，-9 或--best 表示最慢压缩方法(高压缩比)。系统缺省值为 6。

把目录下的每个文件压缩成.gz 文件。
$ gzip *

每个压缩的文件解压，并列出详细的信息。
$ gzip -dv *

详细显示例 1 中每个压缩的文件的信息，并不解压。

$ gzip -l *

压缩一个 tar 备份文件，如 usr.tar，此时压缩文件的扩展名为.tar.gz
$ gzip usr.tar

257 ☆gunzip

对于.gz 结尾的文件
gunzip all.gz

258 ☆bzip2

以.bz2 结尾的文件
bzip2 -d all.bz2
-d 解压

259 ☆bunzip2

bunzip2 all.bz2

260 ☆compress

对一般的文本文件,压缩率较高,可达 81%
实际在 UNIX 下有 zip,gzip 等压缩软件,压缩率比 compress 高,建议使用

使用权限:所有使用者

使用方式:compress [-dfvcV] [-b maxbits] [file ...]

说明:
compress 是一个相当古老的 unix 文件压缩指令，压缩后的文件会加上一个 .Z 延伸档名以区别未压缩的文件，压缩后的文件可以以 uncompress 解压。若要将数个文件压成一个压缩档，必须先将文件 tar 起来再压缩。由于 gzip 可以产生更理想的压缩比例，一般人多已改用 gzip 为文件压缩工具。

参数:
c 输出结果至标准输出设备(一般指荧幕)
f 强迫写入文件，若目的档已经存在，则会被覆盖 (force)
v 将程式执行的讯息印在荧幕上 (verbose)

b 设定共同字串数的上限，以位元计算，可以设定的值为 9 至 16 bits 。由于值越大，能使用的共同字串就 越多，压缩比例就越大，所以一般使用预设值 16 bits (bits)
d 将压缩档解压缩
V 列出版本讯息

将 source.dat 压缩成 source.dat.Z ，若 source.dat.Z 已经存在，内容则会被压缩档覆盖。
compress -f source.dat

将 source.dat 压缩成 source.dat.Z ，并列印出压缩比例。
-v 与 -f 可以一起使用
compress -vf source.dat

将压缩后的资料输出后再导入 target.dat.Z 可以改变压缩档名。
compress -c source.dat > target.dat.Z

-b 的值越大，压缩比例就越大，范围是 9-16 ，预设值是 16 。
compress -b 12 source.dat

将 source.dat.Z 解压成 source.dat ，若文件已经存在，使用者按 y 以确定覆盖文件，若使用 -df 程式则会自动覆盖文件。由于系统会自动加入 .Z 为延伸档名，所以 source.dat 会自动当作 source.dat.Z 处理。
compress -d source.dat
compress -d source.dat.Z

compress /u0/temp1.cpio
compress /u0/temp2.tar
压缩为/u0/temp2.tar.Z
则将文件库/u0/temp1.cpio 压缩为/u0/temp1.cpio.Z(自动添加.Z 并删除/u0/temp1.cpio)

261 ☆uncompress

对于.Z 结尾的文件
uncompress all.Z

uncompress /u0/temp1.cpio.Z
则自动还原为/u0/temp1.cpio

uncompress /u0/temp2.tar.Z
则还原为/u0/temp2.tar

262 ☆zip

zip all.zip *.jpg
这条命令是将所有.jpg 的文件压缩成一个 zip 包

263 ☆unzip

unzip all.zip
这条命令是将 all.zip 中的所有文件解压出来

用 MS Windows 下的压缩软件 winzip 压缩的文件如何在 Linux 系统下展开呢？可以用 unzip 命令，该命令用于解扩展名为.zip 的压缩文件。

语法:unzip [选项] 压缩文件名.zip

各选项的含义分别为:

-x 文件列表 解压缩文件，但不包括指定的 file 文件。
-v 查看压缩文件目录，但不解压。
-t 测试文件有无损坏，但不解压。
-d 目录 把压缩文件解到指定目录下。
-z 只显示压缩文件的注解。
-n 不覆盖已经存在的文件。
-o 覆盖已存在的文件且不要求用户确认。
-j 不重建文档的目录结构，把所有文件解压到同一目录下。

例 1:将压缩文件 text.zip 在当前目录下解压缩。

$ unzip text.zip

例 2:将压缩文件 text.zip 在指定目录/tmp 下解压缩，如果已有相同的文件存在，要求 unzip 命令不覆盖原先的文件。

$ unzip -n text.zip -d /tmp

例 3:查看压缩文件目录，但不解压。
$ unzip -v text.zip

264 ☆rar

要在 linux 下处理.rar 文件，需要安装 RAR for Linux，可以从网上下载，但要记住，RAR for Linux 不是免费的;可从 http://www.rarsoft.com/download.htm 下载 RAR for Linux 3.2.0，然后安装:

tar -xzpvf rarlinux-3.2.0.tar.gz
cd rar
make

rar a all *.jpg
这条命令是将所有.jpg 的文件压缩成一个 rar 包，名为 all.rar，该程序会将.rar 扩展名将自动附加到包名后。

265 ☆unrar

unrar e all.rar
这条命令是将 all.rar 中的所有文件解压出来

266 ☆zgrep

zgrep 命令

这个命令的功能是在压缩文件中寻找匹配的正则表达式，用法和 grep 命令一样，只不过操作的对象是压缩文件。如果用户想看看在某个压缩文件中有没有某一句话，便可用 zgrep 命令。

267 ☆用户和组

root 用户：系统唯一，是真实的，可以登录系统，可以操作系统任何文件和命令，拥有最高权限;
虚拟用户：这类用户也被称之为伪用户或假用户，与真实用户区分开来，这类用户不具有登录系统的能力，但却是系统运行不可缺少的用户，比如 bin、daemon、adm、ftp、mail 等;这类用户都系统自身拥有的，而非后来添加的，当然我们也可以添加虚拟用户;
普通真实用户：这类用户能登录系统，但只能操作自己家目录的内容;权限有限;这类用户都是系统管理员自行添加的;

用户和用户组的对应关系是：一对一、多对一、一对多或多对多;

一对一：某个用户可以是某个组的唯一成员;
多对一：多个用户可以是某个唯一的组的成员，不归属其它用户组;比如 panda 和 linuxsir 两个用户只归属于 panda 用户组;
一对多：某个用户可以是多个用户组的成员;比如 panda 可以是 root 组成员，也可以是 linuxsir 用户组成员，还可以是 adm 用户组成员;
多对多：多个用户对应多个用户组，并且几个用户可以是归属相同的组;其实多对多的关系是前面三条的扩展;理解了上面的三条，这条也能理解;

268☆UID

UID 是用户的 ID 值，在系统中每个用户的 UID 的值是唯一的，更确切的说每个用户都要对应一个唯一的 UID ，系统管理员应该确保这一规则。系统用户的 UID 的值从 0 开始，是一个正整数，至于最大值可以在/etc/login.defs 可以查到，一般 Linux 发行版约定为 60000; 在 Linux 中，root 的 UID 是 0，拥有系统最高权限;

UID 在系统唯一特性，做为系统管理员应该确保这一标准，UID 的唯一性关系到系统的安全，应该值得我们关注！比如我在/etc/passwd 中把 panda 的 UID 改为 0 后，你设想会发生什么呢？panda 这个用户会被确认为 root 用户。panda 这个帐号可以进行所有 root 的操作;

UID 是确认用户权限的标识，用户登录系统所处的角色是通过 UID 来实现的，而非用户名，切记;把几个用户共用一个 UID 是危险的，比如我们上面所谈到的，把普通用户的 UID 改为 0，和 root 共用一个 UID ，这事实上就造成了系统管理权限的混乱。如果我们想用 root 权限，可以通过 su 或 sudo 来实现;切不可随意让一个用户和 root 分享同一个 UID ;

UID 是唯一性，只是要求管理员所做的，其实我们修改/etc/passwd 文件，可以修改任何用户的 UID 的值为 0

一般情况下，每个 Linux 的发行版都会预留一定的 UID 和 GID 给系统虚拟用户占用，虚拟用户一般是系统安装时就有的，是为了完成系统任务所必须的用户，但虚拟用户是不能登录系统的，比如 ftp、nobody、adm、rpm、bin、shutdown 等;

在 Fedora 系统会把前 499 个 UID 和 GID 预留出来，我们添加新用户时的 UID 从 500 开始的，GID 也是从 500 开始，至于其它系统，有的系统可能会把前 999UID 和 GID 预留出来;以各个系统中/etc/login.defs 中的 UID_MIN 的最小值为准; Fedora 系统 login.defs 的 UID_MIN 是 500，而 UID_MAX 值为 60000，也就是说我们通过 adduser 默认添加的用户的 UID 的值是 500 到 60000 之间;而 Slackware 通过 adduser 不指定 UID 来添加用户，默认 UID 是从 1000 开始;

269☆GID

GID 和 UID 类似，是一个正整数或 0，GID 从 0 开始，GID 为 0 的组让系统付予给 root 用户组;系统会预留一些较靠前的 GID 给系统虚拟用户（也被称为伪装用户）之用;每个系统预留的 GID 都有所不同，比如 Fedora 预留了 500 个，我们添加新用户组时，用户组是从 500 开始的;而 Slackware 是把前 100 个 GID 预留，新添加的用户组是从 100 开始;查看系统添加用户组默认的 GID 范围应该查看 /etc/login.defs 中的 GID_MIN 和 GID_MAX 值;

我们可以对照/etc/passwd 和/etc/group 两个文件;我们会发现有默认用户组之说;我们在 /etc/passwd 中的每条用户记录会发现用户默认的 GID ;在/etc/group 中，我们也会发现每个用户组下有多少个用户;在创建目录和文件时，会使用默认的用户组;

但值得注意的是，判断用户的访问权限时，默认的 GID 并不是最重要的，只要一个目录让同组用户可以访问的权限，那么同组用户就可以拥有该目录的访问权，在这时用户的默认 GID 并不是最重要的;

270☆passwd

/etc/passwd 注：用户（user）的配置文件;
在/etc/passwd 中，每一行都表示的是一个用户的信息;一行有 7 个段位;每个段位用:号分割
panda:x:500:500:panda:/home/panda:/bin/bash
第一字段：用户名（也被称为登录名）;
第二字段：口令;在例子中我们看到的是一个 x，其实密码已被映射到/etc/shadow 文件中;
第三字段：UID;
第四字段：GID;
第五字段：用户名全称，这是可选的，可以不设置
第六字段：用户的家目录所在位置;panda 这个用户是/home/panda
第七字段：用户所用 SHELL 的类型，所以设置为/bin/bash ;

271☆shadow

/etc/shadow 注：用户（user）影子口令文件;
/etc/shadow 文件是/etc/passwd 的影子文件，这个文件并不由/etc/passwd 而产生的，这两个文件是应该是对应互补的;shadow 内容包括用户及被加密的密码以及其它/etc/passwd 不能包括的信息，比如用户的有效期限等;这个文件只有 root 权限可以读取和操作

/etc/shadow 的权限不能随便改为其它用户可读，这样做是危险的。如果您发现这个文件的权限变成了其它用户组或用户可读了，要进行检查，以防系统安全问题的发生;

/etc/shadow 文件的内容包括 9 个段位，每个段位之间用:号分割;我们以如下的例子说明;
root:1qIKmQhsa$EO36VQHjjvRz11iFsBPda/:13356:0:99999:7:::
第一字段：用户名（也被称为登录名），在/etc/shadow 中，用户名和/etc/passwd 是相同的，这样就把 passwd 和 shadow 中用的用户记录联系在一起;这个字段是非空的;
第二字段：密码（已被加密），如果是有些用户在这段是 x，表示这个用户不能登录到系统;这个字段是非空的;
第三字段：上次修改口令的时间;这个时间是从 1970 年 01 月 01 日算起到最近一次修改口令的时间间隔（天数），您可以通过 passwd 来修改用户的密码，然后查看/etc/shadow 中此字段的变化;
第四字段：两次修改口令间隔最少的天数;如果设置为 0,则禁用此功能;也就是说用户必须经过多少天才能修改其口令;此项功能用处不是太大;默认值是通过/etc/login.defs 文件定义中获取，PASS_MIN_DAYS 中有定义;
第五字段：两次修改口令间隔最多的天数;这个能增强管理员管理用户口令的时效性，应该说在增强了系统的安全性;如果是系统默认值，是在添加用户时由/etc/login.defs 文件定义中获取，在 PASS_MAX_DAYS 中定义;

第六字段：提前多少天警告用户口令将过期;当用户登录系统后，系统登录程序提醒用户口令将要作废;如果是系统默认值，是在添加用户时由/etc/login.defs 文件定义中获取，在 PASS_WARN_AGE 中定义;
第七字段：在口令过期之后多少天禁用此用户;此字段表示用户口令作废多少天后，系统会禁用此用户，也就是说系统会不能再让此用户登录，也不会提示用户过期，是完全禁用;
第八字段：用户过期日期;此字段指定了用户作废的天数（从 1970 年的 1 月 1 日开始的天数），如果这个字段的值为空，帐号永久可用;
第九字段：保留字段，目前为空，以备将来 Linux 发展之用;

272☆group

/etc/group 注：用户组（group）配置文件;
/etc/group 文件是用户组的配置文件，内容包括用户和用户组，并且能显示出用户是归属哪个用户组或哪几个用户组，因为一个用户可以归属一个或多个不同的用户组;同一用户组的用户之间具有相似的特征。

用户组的特性在系统管理中为系统管理员提供了极大的方便，但安全性也是值得关注的，如某个用户下有对系统管理有最重要的内容，最好让用户拥有独立的用户组，或者是把用户下的文件的权限设置为完全私有;另外 root 用户组一般不要轻易把普通用户加入进去

/etc/group 的内容包括用户组（Group）、用户组口令、GID 及该用户组所包含的用户（User），每个用户组一条记录;格式如下：
group_name:passwd:GID:user_list

在/etc/group 中的每条记录分四个字段：
root:x:0:0:root:/root:/bin/bash
第一字段：用户组名称;
第二字段：用户组密码;x 表示没有设置密码
第三字段：GID
第四字段：用户列表，每个用户之间用,号分割;本字段可以为空;如果字段为空表示用户组为 GID 的用户名;
不包括组的属主

273☆gshadow

/etc/gshadow 注：用户组（group）的影子文件;
/etc/gshadow 是/etc/group 的加密资讯文件，比如用户组（Group）管理密码就是存放在这个文件。/etc/gshadow 和/etc/group 是互补的两个文件;对于大型服务器，针对很多用户和组，定制一些关系结构比较复杂的权限模型，设置用户组密码是极有必要的。比如我们不想让一些非用户组成员永久拥有用户组的权限和特性，这时我们可以通过密码验证的方式来让某些用户临时拥有一些用户组特性，这时就要用到用户组密码;

/etc/gshadow 格式如下，每个用户组独占一行;
groupname:password:admin,admin,...:member,member,...
第一字段：用户组
第二字段：用户组密码，这个段可以是空的或!，如果是空的或有!，表示没有密码;
第三字段：用户组管理者，这个字段也可为空，如果有多个用户组管理者，用,号分割;
第四字段：组成员，如果有多个成员，用,号分割;

274☆skel

/etc/skel 目录一般是存放用户启动文件的目录，这个目录是由 root 权限控制，当我们添加用户时，这个目录下的文件自动复制到新添加的用户的家目录下;/etc/skel 目录下的文件都是隐藏文件，也就是类似.file 格式的;我们可通过修改、添加、删除/etc/skel 目录下的文件，来为用户提供一个统一、标准的、默认的用户环境;

```
[root@panda ~]# ls -la /etc/skel/
总用量 92
drwxr-xr-x 3 root root 4096 8 月  11 23:32 .
drwxr-xr-x 115 root root 12288 10 月  14 13:44 ..
-rw-r--r-- 1 root root 24 5 月  11 00:15 .bash_logout
-rw-r--r-- 1 root root 191 5 月  11 00:15 .bash_profile
-rw-r--r-- 1 root root 124 5 月  11 00:15 .bashrc
-rw-r--r-- 1 root root 5619 2005-03-08 .canna
-rw-r--r-- 1 root root 438 5 月  18 15:23 .emacs
-rw-r--r-- 1 root root 120 5 月  23 05:18 .gtkrc
drwxr-xr-x 3 root root 4096 8 月  11 23:16 .kde
-rw-r--r-- 1 root root 658 2005-01-17 .zshrc
```

/etc/skel 目录下的文件，一般是我们用 useradd 和 adduser 命令添加用户（user）时，系统自动复制到新添加用户（user）的家目录下;如果我们通过修改 /etc/passwd 来添加用户时，我们可以自己创建用户的家目录，然后把/etc/skel 下的文件复制到用户的家目录下，然后要用 chown 来改变新用户家目录的属主;

创建用户时,需要把/etc/skel 目录下的.*隐藏文件复制过去;
cp -R /etc/skel/ /home/panda
ls -la /home/panda/

改变新增用户家目录的属主和权限
ls -ld /home/panda/
发现新增用户的家目录的属主目前是 root ，并且家目录下的隐藏文件也是 root 权限;
chown -R panda:panda /home/panda
确认

ls -ld /home/panda/
ls -la /home/panda/

我们有理由把新增用户家目录的权限设置为只有其自己可读可写可执行
chmod 700 /home/panda/

275 ☆login.defs

/etc/login.defs 文件是当创建用户时的一些规划，比如创建用户时，是否需要家目录，UID 和 GID 的范围;用户的期限等等，这个文件是可以通过 root 来定义的;

比如 Fedora 的 /etc/logins.defs 文件内容;

```
# *REQUIRED*
# Directory where mailboxes reside, _or_ name of file, relative to the
# home directory. If you _do_ define both, MAIL_DIR takes precedence.
# QMAIL_DIR is for Qmail
#
#QMAIL_DIR Maildir
MAIL_DIR /var/spool/mail 注：创建用户时，要在目录/var/spool/mail 中创建一个用户 mail 文件;
#MAIL_FILE .mail
# Password aging controls:
#
# PASS_MAX_DAYS Maximum number of days a password may be used.
# PASS_MIN_DAYS Minimum number of days allowed between password changes.
# PASS_MIN_LEN Minimum acceptable password length.
# PASS_WARN_AGE Number of days warning given before a password expires.
#
PASS_MAX_DAYS 99999 注：用户的密码不过期最多的天数;
PASS_MIN_DAYS 0 注：密码修改之间最小的天数;
PASS_MIN_LEN 5 注：密码最小长度;
PASS_WARN_AGE 7 注：
#
# Min/max values for automatic uid selection in useradd
#
UID_MIN 500 注：最小 UID 为 500 ，也就是说添加用户时，UID 是从 500 开始的;
UID_MAX 60000 注：最大 UID 为 60000;
#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN 500 注：GID 是从 500 开始;
```

GID_MAX 60000
#
If defined, this command is run when removing a user.
It should remove any at/cron/print jobs etc. owned by
the user to be removed (passed as the first argument).
#
#USERDEL_CMD /usr/sbin/userdel_local
#
If useradd should create home directories for users by default
On RH systems, we do. This option is ORed with the -m flag on
useradd command line.
#
CREATE_HOME yes 注：是否创用户家目录，要求创建;

276 ☆newusers

成批添加用户的工具
newusers 后面直接跟一个文件;文件格式和/etc/passwd 的格式相同;
用户名 1:x:UID:GID:用户说明:用户的家目录:所用 SHELL
win00:x:520:520::/home/win00:/sbin/nologin
win01:x:521:521::/home/win01:/sbin/nologin

查看主机上所有 SHELL
chsh --list

277 ☆chpasswd

批量更新用户口令工具
chpasswd 工具是成批更新用户口令的工具，是把一个文件内容重新定向添加到/etc/shadow 中;
chpasswd < 文件
但文件的内容并不是没有约定的，必须以下面的格式来书写，并且不能有空行;
用户名:口令

用户名 1:口令 1
用户名 2:口令 2
win00:123456
win01:654321

最后 pwconv
映射到 /etc/shadow

278 ☆login

功能说明：登入系统。

语法：login

补充说明：login 指令让用户登入系统，您亦可通过它的功能随时更换登入身份。在 Slackware 发行版中 ，您可在指令后面附加欲登入的用户名称，它会直接询问密码，等待用户输入。当/etc 目录里含名称为 nologin 的文件时，系统只 root 帐号登入系统，其他用户一律不准登入。

login 程序负责认证用户(确认用户名和口令相配)，并建立串行线，启动 shell，建立用户的初始环境。

部分初始化设置是输出文件/etc/motd (每天的短信息)的内容，并检查电子邮件。可以在用户家目录中产生一个叫.hushlogin 的文件来是上面所述的失效。

如果存在文件/etc/nologin ，就不允许登录。这个文件一般由 shutdown 及其相关的东西产生。login 检查这个文件，如果这个文件存在，就拒绝接受登录。如果这个文件确实存在，login 就会在退出之前，将它的内容输出到终端。

login 将所有失败的登录企图登记在系统 log 文件中 (通过 syslog)。它也登记所有的 root 的登录。这些都对跟踪入侵者有用。

当前登录着的用户列在/var/run/utmp 中。这个文件直到系统下次启动或关机前有效。系统刚启动时它被清空。它列出了每个用户和用户使用的终端(或网络连接)，及一些有用的信息。 who 、w 及其他类似的命令查看 utmp 文件得到都有谁登录着。

所有成功的登录记录在/var/log/wtmp 中。这个文件将无限制地增大，所以必须有规律的清除，例如有个每周的 cron 任务来清除它。 last 命令浏览 wtmp 文件。

utmp 和 wtmp 都是二进制格 不幸的是，没有特殊的程序(utmpdump)无法查看它们。

279 ☆useradd

功能说明：建立用户帐号。

语法：useradd [-mMnr][-c <备注>][-d <登入目录>][-e <有效期限>][-f <缓冲天数>][-g <群组>][-G <群组>][-s][-u][用户帐号] 或 useradd -D [-b][-e <有效期限>][-f <缓冲天数>][-g <群组>][-G <群组>][-s]

补充说明：useradd 可用来建立用户帐号。帐号建好之后，再用 passwd 设定帐号的密码.而可用 userdel 删除帐号。使用 useradd 指令所建立的帐号，实际上是保存在/etc/passwd 文本文件中。

参数：
-c<备注> 加上备注文字。备注文字会保存在 passwd 的备注栏位中。
-d<登入目录> 指定用户登入时的启始目录。
-D 变更预设值.
-e<有效期限> 指定帐号的有效期限。
-f<缓冲天数> 指定在密码过期后多少天即关闭该帐号。
-g<群组> 指定用户所属的群组。
-G<群组> 指定用户所属的附加群组。
-m 自动建立用户的登入目录。
-M 不要自动建立用户的登入目录。
-n 取消建立以用户名称为名的群组.
-r 建立系统帐号。
-s 指定用户登入后所使用的 shell。
-u 指定用户 ID。

useradd visitor

280☆passwd

使用权限:所有使用者

使用方式:passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]
[username] 指定帐号名称.

说明:用来更改使用者的密码
其中用户名为需要修改口令的用户名。只有超级用户可以使用“passwd 用户名”修改其他用户的口令，普通用户只能用不带参数的 passwd 命令修改自己的口令。
passwd 作为普通用户和超级权限用户都可以运行，但作为普通用户只能更改自己的用户密码，但前提是没有被 root 用户锁定;如果 root 用户运行 passwd ，可以设置或修改任何用户的密码;
passwd 命令后面不接任何参数或用户名，则表示修改当前用户的密码

参数:
Usage: passwd [OPTION...] <accountName>
-k, --keep-tokens keep non-expired authentication tokens 注：保留即将过期的用户在期满后能仍能使用;
-d, --delete delete the password for the named account (root only)注：删除用户密码，仅能以 root 权限操作;
-l, --lock lock the named account (root only)注：锁住用户无权更改其密码，仅能通过 root 权限操作;让某个用户不能修改密码，可以用-l 参数来锁定
-u, --unlock unlock the named account (root only)注：解除锁定;
-f, --force force operation 注：强制操作;仅 root 权限才能操作;

-x, --maximum=DAYS maximum password lifetime (root only) 注：两次密码修正的最大天数，后面接数字;仅能 root 权限操作;
-n, --minimum=DAYS minimum password lifetime (root only) 注：两次密码修改的最小天数，后面接数字，仅能 root 权限操作;
-w, --warning=DAYS number of days warning users receives before password expiration (root only)注：在距多少天提醒用户修改密码;仅能 root 权限操作;
-i, --inactive=DAYS number of days after password expiration when an account becomes disabled (root only)注：在密码过期后多少天，用户被禁掉，仅能以 root 操作;
-S, --status report password status on the named account (root only)注：查询用户的密码状态，仅能 root 用户操作;
--stdin read new tokens from stdin (root only)

passwd -l panda 注：锁定用户 panda 不能更改密码;
passwd -u panda 解锁

passwd -d panda 注：清除 panda 用户密码,当我们清除一个用户的密码时，登录时就无需密码;这一点要加以注意;

passwd -S panda 注：查询 panda 用户密码状态;

passwd panda
Changing password for user panda.
New UNIX password: 注：输入您的密码
Retype new UNIX password: 再输入一次
passwd: all authentication tokens updated successfully.

测试
su panda

281☆chage

chage [-l] [-m 最小天数] [-M 最大天数] [-W 警告] [-I 失效日] [-E 过期日] [-d 最后日] 用户
更改密码有效相关参数

282☆pwconv

修改/etc/passwd 后,让/etc/passwd 和/etc/shadow 同步

283☆pwck

pwck 注：pwck 是校验用户配置文件/etc/passwd 和/etc/shadow 文件内容是否合法或完整;

284 ☆pwunconv

pwunconv 注：是 pwcov 的立逆向操作，是从/etc/shadow 和 /etc/passwd 创建/etc/passwd ，然后会删除 /etc/shadow 文件;

285 ☆useradd

useradd 和 adduser 用法是一样的
useradd 不加参数选项时，后面直接跟所添加的用户名时，系统时读取添加用户配置文件 /etc/login.defs 和/etc/default/useradd 文件，然后读取/etc/login.defs 和/etc/default/useradd 中所定义的规则添加用户;并向/etc/passwd 和/etc/groups 文件添加用户和用户组记录;当然/etc/passwd 和 /etc/groups 的加密资讯文件也同步生成记录;同时发生的还有系统会自动在/etc/add/default 中所约定的目录中建用户的家目录，并复制/etc/skel 中的文件（包括隐藏文件）到新用户的家目录中;

useradd 的语法：
usage: useradd [-u uid [-o]] [-g group] [-G group,...]
[-d home] [-s shell] [-c comment] [-m [-k template]]
[-f inactive] [-e expire] [-p passwd] name
useradd -D [-g group] [-b base] [-s shell]
[-f inactive] [-e expire]

新帐号建立，当不加-D 参数,useradd 指令使用命令列来指定新帐号的设定值并且使用系统上的预设值.新用户帐号将产生一些系统文件，用户目录建立，拷备起始文件等，这些均可以利用命令列选项指定。此版本为 RedHat Linux 提供，可帮每个新加入的用户建立个别的 group,毋须添加-n 选项。

参数:
-c comment 注：新帐号 password 档的说明栏 。这段文字中包括用户真实姓名，办公地址，办公电话等，可以通过 chfn 来更改
-d home_dir 注：新帐号每次登入时所使用的 home_dir。预设值为 default_home 内 login 名称，并当成登入时目录名称。
-e expire_date 注：帐号终止日期。日期的指定格式为 MM/DD/YY。
-f inactive_days 注：帐号过期几日后永久停权。当值为 0 时帐号则立刻被停权。而当值为-1 时则关闭此功能，预设值为-1
-g initial_group 注：group 名称或以数字来做为用户登入起始用户组(group)。用户组名须为现有存在的名称。用户组数字也须为现有存在的用户组。预设的用户组数字为 1。
-G group,[...]
注：定义此用户为此一堆 groups 的成员。每个用户组使用","区格开来，不可以夹杂空白字元。用户组名同-g 选项的限制。定义值为用户的起始用户组。。
-m 注： 用户目录如不存在则自动建立。如使用-k 选项 skeleton_dir 内的文件将复制至用户目录下。然而在/etc/skel 目录下的文件也会复制过去取代。任何在 skeleton_dir or /etc/skel 的目录也相同会在用户目录下一一建立。The-k 同-m 不建立目录以及不复制任何文件为预设值。

-M 不建立用户目录，即使/etc/login.defs 系统档设定要建立用户目录。
-n 预设值用户用户组与用户名称会相同。此选项将取消此预设值。如果不使用-n 参数，系统会自动建一个与用户名同名的用户组;
-r 此参数是用来建立系统帐号。系统帐号的 UID 会比定义在系统档上/etc/login.defs.的 UID_MIN 来的小。注意 useradd 此用法所建立的帐号不会建立用户目录，也不会在乎纪录在/etc/login.defs.的定义值。如果你想要有用户目录须额外指定-m 参数来建立系统帐号。
-s shell 注：用户登入后使用的 shell 名称。预设为不填写，这样系统会帮你指定预设的登入 shell。
-u uid uid 用户的 ID 值。必须为唯一的 ID 值，除非用-o 选项。数字不可为负值。预设为以/etc/login.defs 中的 UID_MIN 的值为准，0 到 UID_MIN 的值之间，为系统保留的 UID ;

useradd -e 11/04/2005 panda 注：添加用户 panda，并设置其有效期为 2005 年 11 月 04 日

useradd -c pandeng -d /opt/panda -G root,users -s /bin/tcsh panda

286 ☆useradd 配置文件

/etc/default/useradd

useradd 加-D 参数后，就是用来改变配置文件 /etc/default/useradd 的;
useradd -D [-g group] [-b base] [-s shell] [-f inactive] [-e expire]

当-D 选项出现时，useradd 秀出现在的预设值，或是藉由命令列的方式更新预设值。可用选项为：
-b default_home 注：定义用户所属目录的前一个目录。用户名称会附加在 default_home 后面用来建立新用户的目录。当然使用-d 后则此选项无效。
-e default_expire_date 注：用户帐号停止日期。
-f default_inactive 注：帐号过期几日后停权。
-g default_group 注：新帐号起始用户组名或 ID。用户组名须为现有存在的名称。用户组 ID 也须为现有存在的用户组。
-s default_shell 注：用户登入后使用的 shell 名称。往后新加入的帐号都将使用此 shell.
如不指定任何参数，useradd 显示目前预设的值。

也可以用编辑器直接操作

useradd -D -s /bin/tcsh 注：把添加用户时的 SHELL 改为 tcsh ;

# useradd defaults file
GROUP=100
HOME=/home 注：把用户的家目录建在/home 中;
INACTIVE=-1 注：是否启用帐号过期停权，-1 表示不启用;
EXPIRE= 注：帐号终止日期，不设置表示不启用;
SHELL=/bin/bash 注：所用 SHELL 的类型;

SKEL=/etc/skel 注： 默认添加用户的目录默认文件存放位置;也就是说，当我们用 adduser 添加用户时，用户家目录下的文件，都是从这个目录中复制过去的;

287 ☆usermod

功能说明：修改用户帐号。

语法：
usermod [-u uid [-o]] [-g group] [-G group,...]
[-d 主目录 [-m]] [-s shell] [-c 注释] [-l 新名称]
[-f 失效日] [-e 过期日] [-p 密码] [-L|-U] 用户名

usermod 不仅能改用户的 SHELL 类型，所归属的用户组，也能改用户密码的有效期，还能改登录名。usermod 如此看来就是能做到用户帐号大转移;
补充说明：usermod 可用来修改用户帐号的各项设定。

参数：
-c comment 更新用户帐号 password 档中的注解栏，一般是使用 chfn(1)来修改。
-d home_dir 更新用户新的登入目录。如果给定-m 选项，用户旧目录会搬到新的目录去，如旧目录不存在则建个新的。
-e expire_date 加上用户帐号停止日期。日期格式为 MM/DD/YY.
-f inactive_days 帐号过期几日后永久停权。当值为 0 时帐号则立刻被停权。而当值为-1 时则关闭此功能。预设值为-1。
-g initial_group 更新用户新的起始登入用户组。用户组名须已存在。用户组 ID 必须参照既有的的用户组。用户组 ID 预设值为 1。
-G group,[...] 定义用户为一堆 groups 的成员。每个用户组使用","区格开来，不可以夹杂空白字元。用户组名同-g 选项的限制。如果用户现在的用户组不再此列，则将用户由该用户组中移除。
-l login_name 变更用户 login 时的名称为 login_name。其它不变。特别是，用户目录名应该也会跟着更动成新的登入名。
-s shell 指定新登入 shell。如此栏留白，系统将选用系统预设 shell。
-u uid 用户 ID 值。必须为唯一的 ID 值，除非用-o 选项。数字不可为负值。预设为最小不得小于/etc/login.defs 中定义的 UID_MIN 值。0 到 UID_MIN 值之间是传统上保留给系统帐号使用。用户目录树下所有的文件目录其 userID 会自动改变。放在用户目录外的文件则要自行手动更动。

警告:usermod 不允许你改变正在线上的用户帐号名称。当 usermod 用来改变 userID,必须确认这名 user 没在电脑上执行任何程序。你需手动更改用户的 crontab 档。也需手动更改用户的 at 工作档。采用 NISserver 须在 server 上更动相关的 NIS 设定。

usermod -d /opt/panda -m -l pandeng -U panda
注：把 panda 用户名改为 pandeng ，并且把其家目录转移到 /opt/panda ;
然后 chown -R fishlinux:fishlinux

usermod -p 123456 panda
警告： usermod 最好不要用它来改用户的密码，因为他在/etc/shadow 中显示的是明口令;修改用户的口令最好用 passwd ;

288 ☆userinfo

userinfo 图形界面的修改工具;
userinfo 系统普通用户都能调用，但都是修改当前操作用户的

289 ☆userdel

功能说明：删除用户帐号。

语法：userdel [-r][用户帐号]

补充说明：userdel 可删除用户帐号与相关的文件。若不加参数，则仅删除用户帐号，而不删除相关文件。

参数：
-r 删除用户登入目录以及目录中所有文件。表示在删除用户的同时，一并把用户的家目录及本地邮件存储的目录或文件也一同删除

userdel panda 注：删除用户 panda，但不删除其家目录及文件;

userdel -r panda 注：删除用户 panda，其家目录及文件一并删除;
请不要轻易用-r 参数;他会删除用户的同时删除用户所有的文件和目录，切记;如果用户目录下有重要的文件，在删除前请备份;

290 ☆userconf

功能说明：用户帐号设置程序。

语法：userconf [--addgroup <群组>][--adduser <用户 ID><群组><用户名称>][--delgroup <群组>][--deluser <用户 ID>][--help]

补充说明：userconf 实际上为 linuxconf 的符号连接，提供图形界面的操作方式，供管理员建立与管理各类帐号。若不加任何参数，即进入图形界面。

参数：
--addgroup<群组> 新增群组。
--adduser<用户 ID><群组><用户名称> 新增用户帐号。
--delgroup<群组> 删除群组。

--deluser<用户 ID> 删除用户帐号。
--help 显示帮助。

用 userdel 删除用户的同时，也会把其用户组删除;

291☆groupadd

语法格式：
groupadd [-g gid [-o]] [-r] [-f] group

groupadd 可指定用户组名称来建立新的用户组帐号，需要时可从系统中取得新用户组值。
groupadd 有下列选项可用。
-g 后接 GID 值，除非使用-o 参数不然该值必须是唯一，不可相同，数值不可为负，预设值以/etc/login.defs 为准;
-r 此参数是用来建立系统帐号的 GID 会比定义在系统档文件上/etc/login.defs 的 GID_MIN 来的小。注意 useradd 此用法所建立的帐号不会建立使用者目录，也不会纪录在/etc/login.defs.的定义值。如果你想要有使用者目录须额外指定-m 参数来建立系统帐号，它会自动帮你选定一个小于的 GID_MIN 的值，不需要再加上-g 参数。
-f This is force flag.新增一个已经存在的用户组帐号，系统会出现错误讯息然后结束 groupadd。如果是这样的情况，不会新增这个用户组(如果是这个情况下，系统不会再新增一次) 也可同时加上-g 选项，当你加上一个 GID，此时 GID 就不用是唯一值，可不加-o 参数，建好用户组后会显结果(adding a group as neither -g or -o options were specified)。

groupadd -g 550 panda

292☆newgrp

功能说明：登入另一个群组。

语法：newgrp [群组名称]

补充说明：newgrp 指令类似 login 指令，当它是以相同的帐号，另一个群组名称，再次登入系统。欲使用 newgrp 指令切换群组，您必须是该群组的用户，否则将无法登入指定的群组。单一用户要同时隶属多个群组，需利用交替用户的设置。若不指定群组名称，则 newgrp 指令会登入该用户名称的预设群组。

293☆gpasswd

设置用户组的密码
gpasswd 用户组

gpasswd linuxsir

正在修改 panda 组的密码
新密码：
请重新输入新密码：

切换组
newgrp panda
密码：

切换后建立文件或组,文件或组所属的组就是切换后的组

294 ☆grpconv

修改/etc/group 后,同步/etc/group 和/etc/gshadow 内容
注：通过/etc/group 和/etc/gshadow 的文件内容来同步或创建/etc/gshadow ，如果/etc/gshadow 不存在则创建;

295 ☆grpck

grpck

296 ☆grpunconv

grpunconv 注：通过/etc/group 和/etc/gshadow 文件内容来同步或创建/etc/group ，然后删除 gshadow 文件;

297 ☆groupmod

功能说明：更改群组识别码或名称。

语法：groupmod [-g <群组识别码> <-o>][-n <新群组名称>][群组名称]

补充说明：需要更改群组的识别码或名称时，可用 groupmod 指令来完成这项工作。

参数：
-g <群组识别码> 设置欲使用的群组识别码。
-o 重复使用群组识别码。
-n <新群组名称> 设置欲使用的群组名称。

298 ☆groupdel

(group delete)

功能说明：删除群组。

语法：groupdel [群组名称]

补充说明：需要从系统上删除群组时，可用 groupdel 指令来完成这项工作。倘若该群组中仍包括某些用户，则必须先删除这些用户后，方能删除群组。

groupdel panda

299 ☆id

功能说明：显示用户的 ID，以及所属群组的 ID。

语法：id [-gGnru][--help][--version][用户名称]

补充说明：id 会显示用户以及所属群组的实际与有效 ID。若两个 ID 相同，则仅显示实际 ID。若仅指定用户名称，则显示目前用户的 ID。

参数：
-g 或--group 显示用户所属群组的 ID。
-G 或--groups 显示用户所属附加群组的 ID。
-n 或--name 显示用户，所属群组或附加群组的名称。
-r 或--real 显示实际 ID。
-u 或--user 显示用户 ID。
-help 显示帮助。
-version 显示版本信息。

在没有加任何参数的情况下，查询的是当前操作用户的用户名、UID 、GID 和所处的主用户组和附属用户组;

300 ☆groups

groups
当前用户属于的组

301 ☆users

users
查看当前登录的用户

302☆who

名称 : who
使用权线 : 所有使用者都可使用

语法:
who [imqsuwHT] [--count] [--idle] [--heading] [--help] [--message] [--mesg] [--version] [--writable] [file] [am i]

说明 : 显示目前登入系统的用户信息。显示系统中有那些使用者正在上面，显示的资料包含了使用者 ID，使用的终端机，从那边连上来的，上线时间，呆滞时间，CPU 使用量，动作等等。
补充说明：执行这项指令可得知目前有那些用户登入系统，单独执行 who 指令会列出登入帐号，使用的终端机，登入时间以及从何处登入或正在使用哪个 X 显示器。
该命令主要用于查看当前在线上的用户情况。这个命令非常有用。如果用户想和其他用户建立即时通讯，比如使用 talk 命令，那么首先要确定的就是该用户确实在线上，不然 talk 进程就无法建立起来。又如，系统管理员希望监视每个登录的用户此时此刻的所作所为，也要使用 who 命令。

参数:
-m 和"who am i"的作用一样，显示运行该程序的用户名。
-q，--count 只显示用户的登录帐号和登录用户的数量，该选项优先级高于其他任何选项。
-s 忽略。主要是用于和其他版本的 who 命令兼容。
-i，-u，--idle 在登录时间后面显示该用户最后一次对系统进行操作至今的时间，也就是常说的"发呆"时间。其中"."符号代表该用户在前 1 秒仍然处于活动状态;"old"则表示该用户空闲已经超过了 24 小时。
-H，--heading 显示一行列标题。常用的标题如表 4-2 所示。
who 命令输出常用标题
标 题 说 明
USER 用户登录帐号
LINE 用户登录使用终端
LOGIN-TIME 用户登录时间
IDLE 用户空闲时间，即未进行操作的时间
PID 用户登录 shell 的进程 ID
FROM 用户网络地址
-w，-T--mesg，--message，--writable 和-s 选项一样，在登录帐号后面显示一个字符来表示用户的信息状态:
+:允许写信息;
-:不允许写信息;
?:不能找到终端设备。
--help 在标准输出上显示帮助信息。
--version 在标准输出上显示版本信息。

所有的选项都是可选的，也就是说可以单独使用 who 命令。不使用任何选项时，who 命令将显示以下三项内容:
login name:登录用户名;
terminal line:使用终端设备;
login time:登录到系统的时间。

如果给出的是两个非选项参数，那么 who 命令将只显示运行 who 程序的用户名、登录终端和登录时间。通常这两个参数是"am i"，即该命令格式为:"who am i"。

如果需要查看在系统上究竟有哪些用户，可以直接使用 who 命令。

查看登录到系统的用户情况
$ who
root pts/1 Jul 20 17:57 (192.168.192.1)
panda :0 Jul 20 17:59
panda pts/2 Jul 20 18:01 (:0.0)

可以看到，现在系统一共有三个用户。第一列是登录用户的帐号;第二列是登录所使用的终端;第三列是登录时间;第四列是用户从什么地方登录的网络地址，这里是域名。

一般来说，这样就可以了解登录用户的大致情况了。但有时上面的显示不是那么直观，因为没有标题说明，不容易看懂，这时就需要使用-H 选项了。

查看登录用户的详细情况，键入:
$ who -uH
NAME LINE TIME IDLE PID COMMENT
root pts/1 Jul 20 17:57 . 3532 (192.168.192.1)
panda :0 Jul 20 17:59 ? 3574
panda pts/2 Jul 20 18:01 00:09 3771 (:0.0)

这样一目了然。其中-u 选项指定显示用户空闲时间，所以可以看到多了一项 IDLE。第一个 root 用户的 IDLE 项是一个".",这就说明该用户在前 1 秒仍然是活动的，而其他用户后面都有一个时间，称为空闲时间。

最后来看看使用"who am i"格式命令的结果:

root pts/1 Jul 20 17:57 (192.168.192.1)

可见只显示出了运行该 who 命令的用户情况，当然这时候不存在空闲时间。

303 ☆whoami

功能说明：先似乎用户名称。

语法：whoami [--help][--version]

补充说明：显示自身的用户名称，本指令相当于执行"id -un"指令。

参数：
--help 在线帮助。
--version 显示版本信息。

304 ☆whois

功能说明：查找并显示用户信息。

语法：whois [帐号名称]

补充说明：whois 指令会去查找并显示指定帐号的用户相关信息，因为它是到 Network Solutions 的 WHOIS 数据库去查找，所以该帐号名称必须在上面注册方能寻获，且名称没有大小写的差别。

305 ☆w

该命令也用于显示登录到系统的用户情况，但是与 who 不同的是，w 命令功能更加强大，它不但可以显示有谁登录到系统，还可以显示出这些用户当前正在进行的工作，并且统计数据相对 who 命令来说更加详细和科学，可以认为 w 命令就是 who 命令的一个增强版。

w 命令的显示项目按以下顺序排列:当前时间，系统启动到现在的时间，登录用户的数目，系统在最近 1 秒、5 秒和 15 秒的平均负载。然后是每个用户的各项数据，项目显示顺序如下:登录帐号、终端名称、远程主机名、登录时间、空闲时间、JCPU、PCPU、当前正在运行进程的命令行。

其中 JCPU 时间指的是和该终端(tty)连接的所有进程占用的时间。这个时间里并不包括过去的后台作业时间，但却包括当前正在运行的后台作业所占用的时间。
而 PCPU 时间则是指当前进程(即在 WHAT 项中显示的进程)所占用的时间。下面介绍该命令的具体用法和参数。

语法:
w -[husfV] [user]

下面对参数进行说明:

-h 不显示各栏位的标题信息列。不显示标题。
-u 忽略执行程序的名称，以及该程序耗费 CPU 时间的信息。当列出当前进程和 CPU 时间时忽略用户名。这主要是用于执行 su 命令后的情况。
-s 使用短模式。不显示用户登入时间，终端机阶段作业和程序所耗费的 CPU 时间。不显示登录时间、JCPU 和 PCPU 时间。
-f 开启或关闭显示用户从何处登入系统。切换显示 FROM 项，也就是远程主机名项。默认值是不显示远程主机名，当然系统管理员可以对源文件作一些修改使得显示该项成为默认值。
-V 显示版本信息。
-l 使用详细格式列表，此为预设值。

User 只显示指定用户的相关情况。

显示当前登录到系统的用户的详细情况
$ w
 18:11:46 up 22 min, 3 users, load average: 0.00, 0.06, 0.10
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root pts/1 192.168.192.1 17:57 0.00s 0.13s 0.02s w
panda :0 - 17:59 ?xdm? 48.17s 1.11s /usr/bin/gnome-session
panda pts/2 :0.0 18:01 10:00 0.02s 0.02s bash

306 ☆logname

功能说明：显示用户名称。

语法：logname [--help][--version]

补充说明：执行 logname 指令，它会显示目前用户的名称。

参数：
--help 在线帮助。
--vesion 显示版本信息。

307 ☆lastb

功能说明：列出登入系统失败的用户相关信息。

语法：lastb [-adRx][-f <记录文件>][-n <显示列数>][帐号名称...][终端机编号...]

补充说明：单独执行 lastb 指令，它会读取位于/var/log 目录下，名称为 btmp 的文件，并把该文件内容
记录的登入失败的用户名单，全部显示出来。

参数：
-a 把从何处登入系统的主机名称或 IP 地址显示在最后一行。
-d 将 IP 地址转换成主机名称。
-f<记录文件> 指定记录文件。
-n<显示列数>或-<显示列数> 设置列出名单的显示列数。
-R 不显示登入系统的主机名称或 IP 地址。
-x 显示系统关机，重新开机，以及执行等级的改变等信息。

308 ☆last

使用权限:所有使用者

语法:
last [options]

说明:
显示系统开机以来获是从每月初登入者的讯息
功能说明：列出目前与过去登入系统的用户相关信息。

语法：last [-adRx][-f <记录文件>][-n <显示列数>][帐号名称...][终端机编号...]

补充说明：单独执行 last 指令，它会读取位于/var/log 目录下，名称为 wtmp 的文件，并把该给文件的内容记录的登入系统的用户名单全部显示出来。

参数：
-a 把从何处登入系统的主机名称或 IP 地址，显示在最后一行。
-d 将 IP 地址转换成主机名称。
-f <记录文件> 指定记录文件。
-n <显示列数>或-<显示列数> 设置列出名单的显示列数。
-R 不显示登入系统的主机名称或 IP 地址。
-x 显示系统关机，重新开机，以及执行等级的改变等信息。

参数:

-R 省略 hostname 的栏位
-num 展示前 num 个
username 展示 username 的登入讯息
tty 限制登入讯息包含终端机代号

```
[root@localhost ~]# last -R -2
root     pts/1        Thu Jul 20 18:43   still logged in
student  tty3         Thu Jul 20 18:43   still logged in
```

wtmp begins Wed Jul 19 22:42:47 2006

[root@localhost ~]# last -2 panda
panda pts/2 :0.0 Thu Jul 20 18:01 - crash (00:40)
panda :0 Thu Jul 20 17:59 - crash (00:42)

wtmp begins Wed Jul 19 22:42:47 2006

309 ☆finger

使用权限: 所有使用者

语法:
finger [options] user[@address]
功能说明：查找并显示用户信息。

语法：finger [-lmsp][帐号名称...]

补充说明：finger 指令会去查找，并显示指定帐号的用户相关信息，包括本地与远端主机的用户皆可，帐号名称没有大小写的差别。单独执行 finger 指令，它会显示本地主机现在所有的用户的登陆信息，包括帐号名称，真实姓名，登入终端机，闲置时间，登入时间以及地址和电话。

参数:
-l 列出该用户的帐号名称，真实姓名，用户专属目录，登入所用的 Shell，登入时间，转信地址，电子邮件状态，还有计划文件和方案文件内容。
-m 禁止对用户真实名字进行匹配;
-s 列出该用户的帐号名称，真实姓名，登入终端机，闲置时间，登入时间以及地址和电话。如果所查询的使用者是远端伺服器的使用者，这个选项无效。
-p 列出该用户的帐号名称，真实姓名，用户专属目录，登入所用的 Shell，登入时间，转信地址，电子邮件状态，但不显示该用户的计划文件和方案文件内容。把.plan 和.project 文件中的内容省略;

说明:finger 可以让使用者查询一些其他使用者的资料。会列出来的资料有:

Login Name
User Name
Home directory
Shell
Login status
mail status
.plan

.project
.forward

其中 .plan ，.project 和 .forward 就是使用者在他的 Home Directory 里的 .plan ， .project 和 .forward 等文件里的资料。如果没有就没有。finger 指令并不限定于在同一伺服器上查询，也可以寻找某一个远端伺服器上的使用者。只要给一个像是 E-mail address 一般的地址即可。

[root@localhost ~]# finger panda
Login: panda Name: panda
Directory: /home/panda Shell: /bin/bash
Last login Thu Jul 20 17:59 (CST) on :0
No mail.
No Plan.

[panda@localhost ~]$ finger -s ftp
Login Name Tty Idle Login Time Office Office Phone
ftp FTP User * * No logins
关于写状态，如果在 Tty 后面 没有任何输出，表示正在写入，如果有*出现，表示没有写入或被禁止，比如下面的例子，ftp 用户没有通过终端登录系统，因为 Tty 是*，同时 Tty 后面还有一个* ，表示禁止写入或没有写入状态(当用户没有登录时）；

310☆chfn

使用权限:所有使用者

语法:
chfn [-f full-name] [-o office] [-p office-phone] [-h home-phone] [-u] [-v] [username]

说明:
主要是用来改用户的全名，办公室地址，电话之类的，用于 finger and mail username

[root@localhost ~]# chfn student
Changing finger information for student.
Name []:
Office []:
Office Phone []:
Home Phone []:

Finger information not changed.

311☆chsh

使用权限:所有使用者

语法:
chsh [-s shell] [--list-shells] [--help] [--version] [username]

说明:
更改使用者 shell 设定
如果 chsh 不加任何参数及用户名的情况下，默认为更改当前操作用户的 SHELL 类型;

[root@localhost ~]# chsh student
Changing shell for student.
New shell [/bin/bash]:

展示 /etc/shells 文件内容
[root@localhost ~]# chsh -l
/bin/sh
/bin/bash
/sbin/nologin
/bin/ash
/bin/bsh
/bin/ksh
/usr/bin/ksh
/usr/bin/pdksh
/bin/tcsh
/bin/csh
/bin/zsh

chsh --list-shells 注：列出当前系统中所有的 SHELL;

chsh -s /bin/ksh panda 注：更改 panda 所用的 shell 为 ksh ;

chsh 还是有用的，特加是不允许用户登录时，我们可以把用户的 SHELL 改到 /sbin/nologin;系统中一些虚拟用户大多是不能登录系统的，这对于系统安全来说是极为重要;

312☆su

(super user)

功能说明：变更用户身份。

语法：su [-flmp][--help][--version][-][-c <指令>][-s][用户帐号]

补充说明：su 可让用户暂时变更登入的身份。变更时须输入所要变更的用户帐号与密码。

参数：
-c<指令>或--command=<指令> 执行完指定的指令后，即恢复原来的身份。
-f 或--fast 适用于 csh 与 tsch，使 shell 不用去读取启动文件。
-,-l 或--login 改变身份时，也同时变更工作目录，以及 HOME,SHELL,USER,LOGNAME。此外，也会变更 PATH 变量。
-m,-p 或--preserve-environment 变更身份时，不要变更环境变量。
-s 或--shell= 指定要执行的 shell。
--help 显示帮助。
--version 显示版本信息。
[用户帐号] 指定要变更的用户。若不指定此参数，则预设变更为 root。

su
su 在不加任何参数，默认为切换到 root 用户，但没有转到 root 用户家目录下，也就是说这时虽然是切换为 root 用户了，但并没有改变 root 登录环境;

su -
su 加参数 - ，表示默认切换到 root 用户，并且改变到 root 用户的环境;

su 参数 - 用户名

su - -c ls
这是 su 的参数组合，表示切换到 root 用户，并且改变到 root 环境，然后列出 root 家目录的文件，然后退出 root 用户;

313☆sudo

功能说明：以其他身份来执行指令。

语法：sudo [-bhHpV][-s][-u <用户>][指令] 或 sudo [-klv]

补充说明：sudo 可让用户以其他的身份来执行指定的指令，预设的身份为 root。在/etc/sudoers 中设置了可执行 sudo 指令的用户。若其未经授权的用户企图使用 sudo，则会发出警告的邮件给管理员。用户使用 sudo 时，必须先输入密码，之后有 5 分钟的有效期限，超过期限则必须重新输入密码。

通过 sudo，我们能把某些超级权限有针对性的下放，并且不需要普通用户知道 root 密码，所以 sudo 相对于权限无限制性的 su 来说，还是比较安全的，所以 sudo 也能被称为受限制的 su ;另外 sudo 是需要授权许可的，所以也被称为授权许可的 su;

参数：
-b 在后台执行指令。
-h 显示帮助。
-H 将 HOME 环境变量设为新身份的 HOME 环境变量。
-k 结束密码的有效期限，也就是下次再执行 sudo 时便需要输入密码。删除时间戳，下一个 sudo 命令要求用求提供密码;
-l 列出用户在主机上可用的和被禁止的命令;一般配置好/etc/sudoers 后，要用这个命令来查看和测试是不是配置正确的;
-p 改变询问密码的提示符号。
-s 执行指定的 shell。
-u<用户> 指定以以某个用户执行特定操作;若不加上此参数，则预设以 root 作为新的身份。
-v 延长密码有效期限 5 分钟。验证用户的时间戳;如果用户运行 sudo 后，输入用户的密码后，在短时间内可以不用输入口令来直接进行 sudo 操作;用-v 可以跟踪最新的时间戳;
-V 显示版本信息。

sudo 执行命令的流程是当前用户切换到 root（或其它指定切换到的用户），然后以 root（或其它指定的切换到的用户）身份执行命令，执行完成后，直接退回到当前用户;而这些的前提是要通过 sudo 的配置文件/etc/sudoers 来进行授权;
但得通过 visudo 来编辑/etc/sudoers 来实现;

sudo -l
来查看哪些命令是可以执行或禁止的;

首先我们通过 visudo 来改/etc/sudoers 文件，加入下面一行;
panda,linuxsir,%panda ALL=/bin/chown,/bin/chmod,/usr/sbin/adduser,/usr/bin/passwd [A-Za-z]*,!/usr/bin/passwd root,/sbin/parted,/sbin/fdisk

然后列出 panda 用户在主机上通过 sudo 可以切换用户所能用的命令或被禁止用的命令;

[panda@localhost ~]$ sudo -l 注：列出用户在主机上能通过切换用户的可用的或被禁止的命令;
Password: 注：在这里输入您的用户密码;
User panda may run the following commands on this host:
(root) /bin/chown 注：可以切换到 root 下用 chown 命令;
(root) /bin/chmod 注：可以切换到 root 下用 chmod 命令;
(root) /usr/sbin/adduser 注：可以切换到 root 下用 adduser 命令;
(root) /usr/bin/passwd [A-Za-z]* 注：可以切换到 root 下用 passwd 命令;
(root) !/usr/bin/passwd root 注：可以切换到 root 下，但不能执行 passwd root 来更改 root 密码;
(root) /sbin/parted 注：可以切换到 root 下执行 parted ;

(root) /sbin/fdisk 注：可以切换到 root 下执行 fdisk ;

通过上面的 sudo -l 列出可用命令后，我想通过 chown 命令来改变/opt 目录的属主为 panda ;

[panda@localhost ~]$ ls -ld /opt 注：查看/opt 的属主;
drwxr-xr-x 26 root root 4096 10 月 27 10:09 /opt 注：得到的答案是归属 root 用户和 root 用户组;
[panda@localhost ~]$ sudo chown panda:panda /opt 注：通过 chown 来改变属主为 panda 用户和 panda 用户组;
[panda@localhost ~]$ ls -ld /opt 注：查看/opt 属主是不是已经改变了;
drwxr-xr-x 26 panda panda 4096 10 月 27 10:09 /opt

我们通过上面的例子发现 panda 用户能切换到 root 后执行改变用户口令的 passwd 命令;但上面的 sudo -l 输出又明文写着不能更改 root 的口令;也就是说除了 root 的口令，panda 用户不能更改外，其它用户的口令都能更改。下面我们来测试;

对于一个普通用户来说，除了更改自身的口令以外，他不能更改其它用户的口令。但如果换到 root 身份执行命令，则可以更改其它用户的口令;

比如在系统中有 linuxsir 这个用户，我们想尝试更改这个用户的口令，

[panda@localhost ~]$ passwd linuxsir 注：不通过 sudo 直接运行 passwd 来更改 linuxsir 用户的口令;
passwd: Only root can specify a user name. 注：失败，提示仅能通过 root 来更改;
[panda@localhost ~]$ sudo passwd linuxsir 注：我们通过/etc/sudoers 的定义，让 panda 切换到 root 下执行 passwd 命令来改变 linuxsir 的口令;
Changing password for user linuxsir.
New UNIX password: 注：输入新口令;
Retype new UNIX password: 注：再输入一次;
passwd: all authentication tokens updated successfully. 注：改变成功;

314 ☆visudo

sudo 的配置文件是/etc/sudoers

visudo 注：visodo 是编辑 /etc/sudoers 的命令;也可以不用这个命令，直接用 vi 来编辑 /etc/sudoers 的效果是一样的;

/etc/sudoers 文件中每行算一个规则，前面带有#号可以当作是说明的内容，并不执行;如果规则很长，一行列不下时，可以用\号来续行，这样看来一个规则也可以拥有多个行;

/etc/sudoers 的规则可分为两类;一类是别名定义，另一类是授权规则;别名定义并不是必须的，但授权规则是必须的;

/etc/sudoers 配置文件中别名规则

别名规则定义格式如下：
Alias_Type NAME = item1, item2, ...
或
Alias_Type NAME = item1, item2, item3 : NAME = item4, item5

Alias_Type
别名类型包括如下四种
Host_Alias 定义主机别名;
User_Alias 用户别名，别名成员可以是用户，用户组（前面要加%号）
Runas_Alias 用来定义 runas 别名，这个别名指定的是“目的用户”，即 sudo 允许切换至的用户;
Cmnd_Alias 定义命令别名;

NAME
就是别名了，NMAE 的命名是包含大写字母、下划线以及数字，但必须以一个大写字母开头，比如 SYNADM、SYN_ADM 或 SYNAD0 是合法的，sYNAMDA 或 1SYNAD 是不合法的;

item
按中文翻译是项目，在这里我们可以译成成员，如果一个别名下有多个成员，成员与成员之间，通过半角,号分隔;成员在必须是有效并事实存在的。什么是有效的呢？比如主机名，可以通过 w 查看用户的主机名（或 ip 地址），如果您只是本地机操作，只通过 hostname 命令就能查看;用户名当然是在系统中存在的，在/etc/paswd 中必须存在;对于定义命令别名，成员也必须在系统中事实存在的文件名（需要绝对路径）;

item 成员受别名类型 Host_Alias、User_Alias、Runas_Alias、Cmnd_Alias 制约，定义什么类型的别名，就要有什么类型的成员相配。我们用 Host_Alias 定义主机别名时，成员必须是与主机相关相关联，比如是主机名（包括远程登录的主机名）、ip 地址（单个或整段）、掩码等;当用户登录时，可以通过 w 命令来查看登录用户主机信息;用 User_Alias 和 Runas_Alias 定义时，必须要用系统用户做为成员;用 Cmnd_Alias 定义执行命令的别名时，必须是系统存在的文件，文件名可以用通配符表示，配置 Cmnd_Alias 时命令需要绝对路径;

其中 Runas_Alias 和 User_Alias 有点相似，但与 User_Alias 绝对不是同一个概念，Runas_Alias 定义的是某个系统用户可以 sudo 切换身份到 Runas_Alias 下的成员;我们在授权规则中以实例进行解说;

别名规则是每行算一个规则，如果一个别名规则一行容不下时，可以通过\来续行;同一类型别名的定义，一次也可以定义几个别名，他们中间用:号分隔，

Host_Alias HT01=localhost,st05,st04,10.0.0.4 255.255.255.0,192.168.1.0/24
定义主机别名 HT01，通过=号列出成员
Host_Alias HT02=st09,st10
主机别名 HT02，有两个成员;
Host_Alias HT01=localhost,st05,st04,10.0.0.4 255.255.255.0,192.168.1.0/24:HT02=st09,st10
上面的两条对主机的定义，可以通过一条来实现，别名之间用:号分割;

注：我们通过 Host_Alias 定义主机别名时，项目可以是主机名、可以是单个 ip（整段 ip 地址也可以），也可以是网络掩码;如果是主机名，必须是多台机器的网络中，而且这些机器得能通过主机名相互通信访问才有效。那什么才算是通过主机名相互通信或访问呢？比如 ping 主机名，或通过远程访问主机名来访问。在我们局域网中，如果让计算机通过主机名访问通信，必须设置/etc/hosts，/etc/resolv.conf ，还要有 DNS 做解析，否则相互之间无法通过主机名访问;在设置主机别名时，如果项目是中某个项目是主机名的话，可以通过 hostname 命令来查看本地主机的主机名，通过 w 命令查来看登录主机是来源，通过来源来确认其它客户机的主机名或 ip 地址;对于主机别名的定义，看上去有点复杂，其实是很简单。

如果您不明白 Host_Alias 是怎么回事，也可以不用设置主机别名，在定义授权规则时通过 ALL 来匹配所有可能出现的主机情况。如果您把主机方面的知识弄的更明白，的确需要多多学习。

User_Alias SYSAD=panda,linuxsir,bnnnb,lanhaitun
定义用户别名，下有四个成员;要在系统中确实在存在的;
User_Alias NETAD=panda,bnnb
定义用户别名 NETAD ，我想让这个别名下的用户来管理网络，所以取了 NETAD 的别名;
User_Alias WEBMASTER=linuxsir
定义用户别名 WEBMASTER，我想用这个别名下的用户来管理网站;
User_Alias SYSAD=panda,linuxsir,bnnnb,lanhaitun:NETAD=panda,bnnb:WEBMASTER=linuxsir
上面三行的别名定义，可以通过这一行来实现，请看前面的说明，是不是符合?

Cmnd_Alias USERMAG=/usr/sbin/adduser,/usr/sbin/userdel,/usr/bin/passwd
[A-Za-z]*,/bin/chown,/bin/chmod
注意：命令别名下的成员必须是文件或目录的绝对路径;
Cmnd_Alias DISKMAG=/sbin/fdisk,/sbin/parted
Cmnd_Alias NETMAG=/sbin/ifconfig,/etc/init.d/network
Cmnd_Alias KILL = /usr/bin/kill
Cmnd_Alias PWMAG = /usr/sbin/reboot,/usr/sbin/halt
Cmnd_Alias SHELLS = /usr/bin/sh, /usr/bin/csh, /usr/bin/ksh, \
/usr/local/bin/tcsh, /usr/bin/rsh, \
/usr/local/bin/zsh
注：这行定义命令别名有点长，可以通过 \ 号断行;
Cmnd_Alias SU = /usr/bin/su,/bin,/sbin,/usr/sbin,/usr/bin

在上面的例子中，有 KILL 和 PWMAG 的命令别名定义，我们可以合并为一行来写，也就是等价行;

Cmnd_Alias KILL = /usr/bin/kill:PWMAG = /usr/sbin/reboot,/usr/sbin/halt
这一行就代表了 KILL 和 PWMAG 命令别名，把 KILL 和 PWMAG 的别名定义合并在一行写也是可以的;

Runas_Alias OP = root, operator
Runas_Alias DBADM=mysql:OP = root, operator
这行是上面两行的等价行;至于怎么理解 Runas_Alias ，我们必须得通过授权规则的实例来理解;

/etc/sudoers 中的授权规则：

授权规则是分配权限的执行规则，我们前面所讲到的定义别名主要是为了更方便的授权引用别名;如果系统中只有几个用户，其实下放权限比较有限的话，可以不用定义别名，而是针对系统用户直接直接授权，所以在授权规则中别名并不是必须的;

如果您想详细了解授权规则写法的，请参看 man sudoers

授权用户 主机=命令动作

这三个要素缺一不可，但在动作之前也可以指定切换到特定用户下，在这里指定切换的用户要用( )号括起来，如果不需要密码直接运行命令的，应该加 NOPASSWD:参数，但这些可以省略;举例说明;

panda ALL=/bin/chown,/bin/chmod
如果我们在/etc/sudoers 中添加这一行，表示 panda 可以在任何可能出现的主机名的系统中，可以切换到 root 用户下执行 /bin/chown 和/bin/chmod 命令，通过 sudo -l 来查看 panda 在这台主机上允许和禁止运行的命令;

值得注意的是，在这里省略了指定切换到哪个用户下执行/bin/shown 和/bin/chmod 命令;在省略的情况下默认为是切换到 root 用户下执行;同时也省略了是不是需要 panda 用户输入验证密码，如果省略了，默认为是需要验证密码。

为了更详细的说明这些，我们可以构造一个更复杂一点的公式;

授权用户 主机=[(切换到哪些用户或用户组)] [是否需要密码验证] 命令 1,[(切换到哪些用户或用户组)] [是否需要密码验证] [命令 2],[(切换到哪些用户或用户组)] [是否需要密码验证] [命令 3].....

注解：
凡是[ ]中的内容，是可以省略;命令与命令之间用,号分隔;通过本文的例子，可以对照着看哪些是省略了，哪些地方需要有空格;

在[(切换到哪些用户或用户组)] ，如果省略，则默认为 root 用户;如果是 ALL ，则代表能切换到所有用户;注意要切换到的目的用户必须用()号括起来，比如(ALL)、(panda)

panda ALL=(root) /bin/chown, /bin/chmod
表示的是 panda 可以在任何可能出现的主机名的主机中，可以切换到 root 下执行 /bin/chown ，可以切换到任何用户招执行/bin/chmod 命令，通过 sudo -l 来查看 panda 在这台主机上允许和禁止运行的命令;

panda ALL=(root) NOPASSWD: /bin/chown,/bin/chmod
表示的是 panda 可以在任何可能出现的主机名的主机中，可以切换到 root 下执行 /bin/chown ，不需要输入 panda 用户的密码;并且可以切换到任何用户下执行/bin/chmod 命令，但执行 chmod 时需要 panda 输入自己的密码;通过 sudo -l 来查看 panda 在这台主机上允许和禁止运行的命令;

关于一个命令动作是不是需要密码，我们可以发现在系统在默认的情况下是需要用户密码的，除非特加指出不需要用户需要输入自己密码，所以要在执行动作之前加入 NOPASSWD: 参数;

比如我们想用 panda 普通用户通过 more /etc/shadow 文件的内容时，可能会出现下面的情况;
[panda@localhost ~]$ more /etc/shadow
/etc/shadow: 权限不够

这时我们可以用 sudo more /etc/shadow 来读取文件的内容;就就需要在/etc/soduers 中给 panda 授权;

于是我们就可以先 su 到 root 用户下通过 visudo 来改/etc/sudoers ;（比如我们是以 panda 用户登录系统的）
[panda@localhost ~]$ su
Password: 注：在这里输入 root 密码
下面运行 visodu;
[root@panda ~]# visudo
运行 visudo 来改 /etc/sudoers

加入如下一行，退出保存;退出保存，在这里要会用 vi，visudo 也是用的 vi 编辑器;至于 vi 的用法不多说了;
panda ALL=/bin/more 表示 panda 可以切换到 root 下执行 more 来查看文件;

退回到 panda 用户下，用 exit 命令;
[root@panda ~]# exit
exit
[panda@localhost ~]$

查看 panda 的通过 sudo 能执行哪些命令？

[panda@localhost ~]$ sudo -l

Password: 注：在这里输入 panda 用户的密码
User panda may run the following commands on this host: 注：在这里清晰的说明在本台主机上，panda 用户可以以 root 权限运行 more ;在 root 权限下的 more ，可以查看任何文本文件的内容的;
    (root) /bin/more

最后，我们看看是不是 panda 用户有能力看到/etc/shadow 文件的内容;
[panda@localhost ~]$ sudo more /etc/shadow

panda 不但能看到 /etc/shadow 文件的内容，还能看到只有 root 权限下才能看到的其它文件的内容，比如;
[panda@localhost ~]$ sudo more /etc/gshadow

对于 panda 用户查看和读取所有系统文件中，我只想把/etc/shadow 的内容可以让他查看;可以加入下面的一行;
panda ALL=/bin/more /etc/shadow

如果用户组出现在/etc/sudoers 中，前面要加%号，比如%panda ，中间不能有空格;

%panda ALL=/usr/sbin/*,/sbin/*
如果我们在 /etc/sudoers 中加上如上一行，表示 panda 用户组下的所有成员，在所有可能的出现的主机名下，都能切换到 root 用户下运行 /usr/sbin 和/sbin 目录下的所有命令;

取消程序某类程序的执行，要在命令动作前面加上!号; 在本例中也出现了通配符的*的用法;

panda ALL=/usr/sbin/*,/sbin/*,!/usr/sbin/fdisk 注：把这行规则加入到/etc/sudoers 中;但您得有 panda 这个用户组，并且 panda 也是这个组中的才行;

本规则表示 panda 用户在所有可能存在的主机名的主机上运行/usr/sbin 和/sbin 下所有的程序，但 fdisk 程序除外;

[panda@localhost ~]$ sudo -l
Password: 注：在这里输入 panda 用户的密码;
User panda may run the following commands on this host:
(root) /usr/sbin/*
(root) /sbin/*
(root) !/sbin/fdisk

[panda@localhost ~]$ sudo /sbin/fdisk -l
Sorry, user panda is not allowed to execute '/sbin/fdisk -l' as root on localhost.
注：不能切换到 root 用户下运行 fdisk 程序;

别名的运用的实践;

假如我们就一台主机 localhost，能通过 hostname 来查看，我们在这里就不定义主机别名了，用 ALL 来匹配所有可能出现的主机名;并且有 panda、linuxsir、lanhaitun 用户;主要是通过小例子能更好理解;sudo 虽然简单好用，但能把说的明白的确是件难事;最好的办法是多看例子和 man soduers ;

User_Alias SYSADER=panda,linuxsir,%panda
User_Alias DISKADER=lanhaitun
Runas_Alias OP=root
Cmnd_Alias SYDCMD=/bin/chown,/bin/chmod,/usr/sbin/adduser,/usr/bin/passwd [A-Za-z]*,!/usr/bin/passwd root
Cmnd_Alias DSKCMD=/sbin/parted,/sbin/fdisk 注：定义命令别名 DSKCMD，下有成员 parted 和 fdisk ;
SYSADER ALL= SYDCMD,DSKCMD
DISKADER ALL=(OP) DSKCMD

注解：
第一行：定义用户别名 SYSADER 下有成员 panda、linuxsir 和 panda 用户组下的成员，用户组前面必须加%号;
第二行：定义用户别名 DISKADER ，成员有 lanhaitun
第三行：定义 Runas 用户，也就是目标用户的别名为 OP，下有成员 root
第四行：定义 SYSCMD 命令别名，成员之间用,号分隔，最后的!/usr/bin/passwd root 表示不能通过 passwd 来更改 root 密码;
第五行：定义命令别名 DSKCMD，下有成员 parted 和 fdisk ;
第六行： 表示授权 SYSADER 下的所有成员，在所有可能存在的主机名的主机下运行或禁止 SYDCMD 和 DSKCMD 下定义的命令。更为明确遥说，panda、linuxsir 和 panda 用户组下的成员能以 root 身份运行 chown 、chmod 、adduser、passwd，但不能更改 root 的密码;也可以以 root 身份运行 parted 和 fdisk ，本条规则的等价规则是;

panda,linuxsir,%panda ALL=/bin/chown,/bin/chmod,/usr/sbin/adduser,/usr/bin/passwd [A-Za-z]*,!/usr/bin/passwd root,/sbin/parted,/sbin/fdisk

第七行：表示授权 DISKADER 下的所有成员，能以 OP 的身份，来运行 DSKCMD ，不需要密码;更为明确的说 lanhaitun 能以 root 身份运行 parted 和 fdisk 命令;其等价规则是：

lanhaitun ALL=(root) /sbin/parted,/sbin/fdisk

可能有的弟兄会说我想不输入用户的密码就能切换到 root 并运行 SYDCMD 和 DSKCMD 下的命令，那应该把把 NOPASSWD:加在哪里为好？理解下面的例子吧，能明白的;

SYSADER ALL= NOPASSWD: SYDCMD, NOPASSWD: DSKCMD

/etc/sudoers 中其它的未尽事项;

在授权规则中，还有 NOEXEC:和 EXEC 的用法，自己查 man sudoers 了解;还有关于在规则中通配符的用法，也是需要了解的。这些内容不多说了，毕竟只是一个入门性的文档。soduers 配置文件要多简单就有多简单，要多难就有多难，就看自己的应用了。

315 ☆sudoedit

sudoedit 注：和 sudo 功能差不多;

316 ☆sliplogin

功能说明：将 SLIP 接口加入标准输入。

语法：sliplogin [用户名称]

补充说明：sliplogin 可将 SLIP 接口加入标准输入，把一般终端机的连线变成 SLIP 连线。通常可用来建立 SLIP 服务器，让远端电脑以 SLIP 连线到服务器。sliplogin 活去检查/etc/slip/slip.hosts 文件中是否有相同的用户名称。通过检查后，sliplogin 会调用执行 shell script 来设置 IP 地址，子网掩码等网络界面环境。此 shell script 通常是/etc/slip/slip.login。

317 ☆logout

功能说明：退出系统。

语法：logout

补充说明：logout 指令让用户退出系统，其功能和 login 指令相互对应。

318 ☆vlock

(virtual console lock)

功能说明：锁住虚拟终端。

语法：vlock [-achv]

补充说明：执行 vlock 指令可锁住虚拟终端，避免他人使用。

参数：
-a 或--all 锁住所有的终端阶段作业，如果您在全屏幕的终端中使用本参数，则会将用键盘

切换终端机的功能一并关闭。
-c 或--current 锁住目前的终端阶段作业，此为预设值。
-h 或--help 在线帮助。
-v 或--version 显示版本信息。

319 ☆sync

sync 命令是在关闭 Linux 系统时使用的。

一般正常的关闭系统的过程是自动进行这些工作的，在系统运行过程中也会定时做这些工作，不需要用户干预。

sync 命令是强制把内存中的数据写回硬盘，以免数据的丢失。用户可以在需要的时候使用此命令。该命令的一般格式为：
sync

320 ☆shutdown

语法：shutdown [-efFhknr][-t 秒数][时间][警告信息]
执行 shutdown 命令时，系统会通知所有登录的用户系统将要关闭，并且 login 指令会被冻结，即新的用户不能再登录系统。使用 shutdown 命令可以直接关闭系统，也可以延迟指定的时间再关闭系统，还可以重新启动。延迟指定的时间再关闭系统，可以让用户有时间储存当前正在处理的文件和关闭已经打开的程序。

参数：
-t 指定在多长时间之后关闭系统
-k 并不真正关机，而只是发出警告信息给所有用户。
-r 关机后立即重新启动。
-h 关机后不重新启动。
-f 快速关机，重启动时跳过 fsck。
-n 快速关机，不经过 init 程序。
-c 取消一个已经运行的 shutdown。只要按+键就可以中断关机的指令。
-f 重新启动时不执行 fsck。
-n 不调用 init 程序进行关机，而由 shutdown 自己进行。
[时间] 设置多久时间后执行 shutdown 指令。
[警告信息] 要传送给所有登入用户的信息。

时间参数，可以是一个精确的时间，也可以是从现在开始的一个时间段。精确时间的格式是 hh:mm，表示小时和分钟;时间段由“+”和分钟数表示。系统执行该命令后，会自动进行数据同步的工作。

shutdown 命令的工作实质是给 init 程序发送信号（signal），要求其切换系统的运行级别（Runlevel）。系统的运行级别包括：
0：关闭系统
1：单用户模式，如果没有为 shutdown 命令指定-h 或-r 参数而直接执行，则默认将切换到此运行级别
2：多用户模式（不支持 NFS）
3：多用户模式（支持 NFS），一般常用此种运行级别
5：多用户模式（GUI 模式）
6：重新启动系统

补充说明：shutdown 指令可以关闭所有程序，并依用户的需要，进行重新开机或关机的动作。

关机
init 同样也用来控制系统关机或者重新启动，通过 shutdown 命令可以实现关机操作。要立即关闭计算机，可以使用 shutdown -h now 命令，h 代表的意思就是 halt，也就是切断了电源，如果是重新启动，直接把 h 参数换成 r 就可以了：shutdown -r now
关机过程需要一些时间，这个过程中不应该手工切断电源或者按下 reset 键。上面的例子里 now 参数表示的就是立即的意思，也有许多其他的参数可以选用，比如+n 设定倒计时时间，n 就是你想要的时间，这些都可以通过 man shutdown 了解一下。
举例说明，要使系统 10 分钟后重新起动：shutdown -r +10
在 linux 系统里，shutdown 命令会通知已经登录进来的用户它即将关机，不过意义不大罢了。如果定义了倒计时关机，shutdown 命令会生成一个/etc/nologin 文件，这个文件的存在能够阻止其他用户再进行登录操作，当然了，root 用户除外。
在系统关机时，shutdown 命令会告诉 init 程序转换到 0 运行级别，如果是重新启动则转换到第 6 运行级别。当进入 0 或者 6 运行级别后，系统将会依照下面的顺序运行：

1,init 关闭所有它能关闭的进程(转换到其他运行级别也一样)
2,rc0.d/rc6.d 目录下的第一个命令开始运行，锁定系统文件为关机作准备
3,rc0.d/rc6.d 目录下的第二个命令运行，卸载除根文件系统以外的所有文件系统(如挂载的 windows 分区)
4,rc0.d/rc6.d 中的命令将把根文件系统重新挂载为只读属性
5,rc0.d/rc6.d 中的命令调用 sync 程序把缓存中的数据写入文件系统
6,最后的命令是重新启动或者关闭内核程序

321 ☆halt

halt 是最简单的关机命令，其实际上是调用 shutdown -h 命令。halt 执行时，杀死应用进程，文件系统写操作完成后就会停止内核。
halt 命令的部分参数如下：
[-f] 没有调用 shutdown 而强制关机或重启
[-i] 关机或重新启动之前，关掉所有的网络接口
[-p] 关机时调用 poweroff，此选项为缺省选项

功能说明：关闭系统。

语法：halt [-dfinpw]

补充说明：halt 会先检测系统的 runlevel。若 runlevel 为 0 或 6，则关闭系统，否则即调用 shutdown 来关闭系统。

参数：
-d 不要在 wtmp 中记录。
-f 不论目前的 runlevel 为何，不调用 shutdown 即强制关闭系统。
-i 在 halt 之前，关闭全部的网络界面。
-n halt 前，不用先执行 sync。
-p halt 之后，执行 poweroff。
-w 仅在 wtmp 中记录，而不实际结束系统。

322 ☆reboot

功能说明：重新开机。

语法：reboot [-dfinw]

补充说明：执行 reboot 指令可让系统停止运作，并重新开机。

参数：
-d 重新开机时不把数据写入记录文件/var/tmp/wtmp。本参数具有"-n"参数的效果。
-f 强制重新开机，不调用 shutdown 指令的功能。
-i 在重开机之前，先关闭所有网络界面。
-n 重开机之前不检查是否有未结束的程序。
-w 仅做测试，并不真的将系统重新开机，只会把重开机的数据写入/var/log 目录下的 wtmp 记录文件。

reboot 的工作过程与 halt 类似，其作用是重新启动，而 halt 是关机。其参数也与 halt 类似。

323 ☆init

init 是所有进程的祖先，其进程号始终为 1。init 用于切换系统的运行级别，切换的工作是立即完成的。init 0 命令用于立即将系统运行级别切换为 0，即关机;init 6 命令用于将系统运行级别切换为 6，即重新启动。

324 ☆uname

功能说明：显示系统信息。

语法：uname [-amnrsv][--help][--version]

说明：uname 可显示电脑以及操作系统的相关信息。

参数：
-a 或--all 显示全部的信息。
-m 或--machine 显示电脑类型。
-n 或-nodename 显示在网络上的主机名称。
-r 或--release 显示操作系统的发行编号。
-s 或--sysname 显示操作系统名称。
-v 显示操作系统的版本。
--help 显示帮助。
--version 显示版本信息。

325 ☆uptime

使用权限: 所有使用者
使用方式: uptime [-V]

说明:
uptime 命令显示系统已经运行了多长时间，它依次显示下列信息：现在时间、系统已经运行了多长时间、目前有多少登录用户、系统在过去的 1 分钟、5 分钟和 15 分钟内的平均负载。

参数:
-V 显示版本资讯。

uptime 提供使用者下面的资讯，不需其他参数:
现在的时间
系统开机运转到现在经过的时间
连线的使用者数量
最近一分钟，五分钟和十五分钟的系统负载

uptime
10:41am up 5 days, 10 min, 1 users, load average: 0.00, 0.00, 1.99

326 ☆free

语法： free [-bkmotV][-s <间隔秒数>]

功能说明：free 命令的功能是查看当前系统内存的使用情况，它显示系统中剩余及已用的物理内存和交换内存，以及共享内存和被核心使用的缓冲区。

参数：
-b 以 Byte 为单位显示内存使用情况。
-k 以 KB 为单位显示内存使用情况。
-m 以 MB 为单位显示内存使用情况。
-o 不显示缓冲区调节列。
-s<间隔秒数> 持续观察内存使用状况。
-t 显示内存总和列。
-V 显示版本信息。

327 ☆logrotate

功能说明：管理记录文件。

语法：logrotate [-?dfv][-s <状态文件>][--usage][配置文件]

补充说明：使用 logrotate 指令，可让你轻松管理系统所产生的记录文件。它提供自动替换，压缩，删除和邮寄记录文件，每个记录文件都可被设置成每日，每周或每月处理，也能在文件太大时立即处理。您必须自行编辑，指定配置文件，预设的配置文件存放在/etc 目录下，文件名称为 logrotate.conf。

参数：
-?或--help 在线帮助。
-d 或--debug 详细显示指令执行过程，便于排错或了解程序执行的情况。
-f 或--force 强行启动记录文件维护操作，纵使 logrotate 指令认为没有需要亦然。
-s<状态文件>或--state=<状态文件> 使用指定的状态文件。
-v 或--version 显示指令执行过程。
-usage 显示指令基本用法。

328 ☆作业

一个正在执行的进程称为一个作业，而且作业可以包含一个或多个进程，尤其是当使用了管道和重定向命令。例如“nroff -man ps.1|grep kill|more”这个作业就同时启动了三个进程。
这时候实际上是同时启动了三个进程。请注意是同时启动的，所有放在管道两边的进程都将被同时启动，它们都是当前 shell 的子程序，互相之间可以称为兄弟进程。

329 ☆程序

程序是为了完成某种任务而设计的软件
程序只是一个静态的指令集合，不占系统的运行资源;

330 ☆进程

进程就是运行中的程序,在自身的虚拟地址空间运行的一个单独的程序。
进程是一个随时都可能发生变化的、动态的、使用系统运行资源的程序。而且一个程序可以启动多个进程。

进程分类:
进程一般分为交互进程、批处理进程和守护进程三类。

交互进程--由一个 shell 启动的进程。交互进程既可以在前台运行，也可以在后台运行。
批处理进程--这种进程和终端没有联系，是一个进程序列。
监控进程（也称守护进程）--Linux 系统启动时启动的进程，并在后台运行。

进程的属性:
进程 ID（PID)：是唯一的数值，用来区分进程;
父进程和父进程的 ID（PPID);
启动进程的用户 ID（UID）和所归属的组（GID）;
进程状态：状态分为运行 R、休眠 S、僵尸 Z;
进程执行的优先级;
进程所连接的终端名;
进程资源占用：比如占用资源大小（内存、CPU 占用量）;

父进程和子进程:
当父进程终止时，子进程也随之而终止。但子进程终止，父进程并不一定终止。

331 ☆&

语法: 命令&
例如:
cc file1.c & 将编译 file1.c 文件的工作置于后台执行

语法:按下 Control+Z 健,暂停正在执行的进程.键入 bg 命令,将暂停的进程置于后台继续执行.
例如:
cc file2.c
^Z
Stopped
bg

进程的挂起及恢复命令 bg、fg

作业控制允许将进程挂起并可以在需要时恢复进程的运行，被挂起的作业恢复后将从中止处开始继续运行。只要在键盘上按<ctrl+z>，即可挂起当前的前台作业。

恢复进程执行时，有两种选择：用 fg 命令将挂起的作业放回到前台执行;用 bg 命令将挂起的作业放到后台执行。

默认情况下，fg 和 bg 命令对最近停止的作业进行操作。如果希望恢复其他作业的运行，可以在命令中指定要恢复作业的作业号来恢复该作业。

332 ☆jobs

语法:jobs

333 ☆nohup

理论上，我们一般退出 Linux 系统时，会把所有的程序全部结束掉，包括那些后台程序。但有时候，例如您正在编辑一个很长的程序，但是您下班或是有事需要先退出系统，这时您又不希望系统把您编辑那么久的程序结束掉，希望退出系统时，程序还能继续执行。这时，我们就可以使用 nohup 命令使进程在用户退出后仍继续执行。

一般这些进程我们都是让它在后台执行，结果则会写到用户自己的目录下的 nohup.out 这个文件里(也可以使用输出重定向，让它输出到一个特定的文件)。

$ nohup sort sales.dat &

这条命令告诉 sort 命令忽略用户已退出系统，它应该一直运行，直到进程完成。利用这种方法，可以启动一个要运行几天甚至几周的进程，而且在它运行时，用户不需要去登录。

nohup 命令把一条命令的所有输出和错误信息送到 nohup.out 文件中。若将输出重定向，则只有错误信息放在 nohup.out 文件中。

可以使用 nohup run.sh & 这个命令。

如果你是使用客户机远程登录到主机上的，那么退出 ssh 和关闭你当前的客户机都是没有问题的，要是你在客户机上运行的话，关机当然是不行的啦，呵呵

334 ☆screen

功能说明：多重视窗管理程序。

语法：screen [-AmRvx -ls -wipe][-d <作业名称>][-h <行数>][-r <作业名称>][-s][-S <作业名称>]

补充说明：screen 为多重视窗管理程序。此处所谓的视窗，是指一个全屏幕的文字模式画面。通常只有在使用 telnet 登入主机或是使用老式的终端机时，才有可能用到 screen 程序。

参数：
-A 将所有的视窗都调整为目前终端机的大小。
-d<作业名称> 将指定的 screen 作业离线。
-h<行数> 指定视窗的缓冲区行数。
-m 即使目前已在作业中的 screen 作业，仍强制建立新的 screen 作业。
-r<作业名称> 恢复离线的 screen 作业。
-R 先试图恢复离线的作业。若找不到离线的作业，即建立新的 screen 作业。
-s 指定建立新视窗时，所要执行的 shell。
-S<作业名称> 指定 screen 作业的名称。
-v 显示版本信息。
-x 恢复之前离线的 screen 作业。
-ls 或--list 显示目前所有的 screen 作业。
-wipe 检查目前所有的 screen 作业，并删除已经无法使用的 screen 作业。

虚拟终端
Vincent Danen 为我们举例说明如何应用 Linux 工具 screen 建立虚拟终端。

每个系统管理员都熟悉用 SSH 进行远程管理;同样，同时做多项工作，包括冗长的编译或长期任务，也是我们经常做的工作。通常，释放控制台要打开新的终端，并建立一个新的 SSH 连接以完成其它工作;或者--如果任务观测不是很重要的话--也可以将任务送交后台运行来释放终端。

另一个方法就是应用 screen 工具--包含在所有 Linux 产品中的一个程序。screen 建立一个你能够控制并可通过一个终端交互的虚拟终端。更好的是，你不必中断一个运行中的任务就可终止一个 screen 会话。想象一下，在一个远程服务器开始一个冗长编译工作，但连接中断;连接中断时，你的任务也中断了。Screen 允许你与运行的会话分离、登录出去，随后再（甚至是从一个不同的地方）恢复它，从而避免这一问题。

首先，确认通过软件包管理器安装了 screen 包，然后输入：
$ screen

这样就启动 screen 并打开一个新会话。要断开会话，输入 CTRL-A，接着再输入 d，你就会返回你启动 screen 的提示符，但你在 screen 中所做的工作依然有效。如果只有一个 screen 会话在运行之中，你可以这样连接它：
$ screen -R

如果有几个 screen 会话在运行，这种方法就连接不上 screen 会话。但你可以用下面的方法来查看运行中的 screen 会话：

$ screen -list
There are screens on:
13995.pts-0.host(Detached)

14529.pts-0.host(Attached)
2 Sockets in /home/joe/tmp.

从这里可以看到，有两个会话正在运行。要从不同的地点连接分离的会话，可以应用（当然，要在机器上应用 SSH）：
$ screen -r 13995
这里的 13995 是你希望连接上的 screen 会话程序 ID。

Screen 有许多有效的帮助信息，你可以用它做许多事情。你可以查看 screen 使用说明，screen -help 的输出结果，并在命令模式下的 screen 会话内，输入 CTRL-A 与？来获得你能应用的命令列表（由 CTRL-A 调用）。

335 ☆at

使用权限 ：所有使用者

语法 ：
at -V [-q queue] [-f file] [-mldbv] TIME
at -c 作业 [作业...]

说明 : at 可以让使用者指定在 TIME 这个特定时刻执行某个程式或指令，TIME 的格式是当天的 hh:mm（小时:分钟）式的时间指定。如果该时间已经过去，那么就放在第二天执行。当然也可以使用 midnight（深夜），noon（中午），teatime（饮茶时间，一般是下午 4 点）等比较模糊的词语来指定时间。用户还可以采用 12 小时计时制，即在时间后面加上 AM（上午）或者 PM（下午）来说明是上午还是下午。

如果想要指定超过一天内的时间，指定格式为 month day（月 日）或者 mm/dd/yy（月/日/年）或者 dd.mm.yy（日.月.年）。指定的日期必须跟在指定时间的后面。

相对计时法，这对于安排不久就要执行的命令是很有好处的。指定格式为：now + count time-units ，now 就是当前时间，time-units 是时间单位，这里可以是 minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count 是时间的数量，究竟是几天，还是几小时，等等。

另外，使用者也可指定 today 或 tomorrow 来表示今天或明天。当指定了时间并按下 enter 之后，at 会进入交谈模式并要求输入指令或程式，当你输入完后按下 ctrl+D 即可完成所有动作，至于执行的结果将会寄回你的帐号中。

/var/spool/at 目录中存储记录

在任何情况下，超级用户都可以使用这个命令。对于其他用户来说，是否可以使用就取决于两个文件：/etc/at.allow 和/etc/at.deny。
如果/etc/at.allow 文件存在的话，那么只有在其中列出的用户才可以使用 at 命令;

如果该文件不存在，那么将检查/etc/at.deny 文件是否存在，在这个文件中列出的用户均不能使用该命令。
如果两个文件都不存在，那么只有超级用户可以使用该命令;
空的/etc/at.deny 文件意味着所有的用户都可以使用该命令，这也是默认状态。

<Ctrl+d>组合键结束 at 命令的输入.

参数 :

-V : 印出版本编号
-q queue 使用指定的队列。队列名称是由单个字母组成，合法的队列名可以由 a-z 或者 A-Z。a 队列是 at 命令的默认队列。使用者可以同时使用多个 queue
-m : 即使程式/指令执行完成后没有输出结果, 也要发送邮件给使用者
-f file : 读入预先写好的命令文件，而不是从标准输入读取。使用者不一定要使用交谈模式来输入，可以先将所有的指定先写入文件后再一次读入
-l : 该命令用于查看安排的作业序列，它将列出用户排在队列中的作业，如果是超级用户，则列出队列中的所有工作。(使用者也可以直接使用 atq 而不用 at -l)
-d : 该命令用于删除指定要执行的命令序列(使用者也可以直接使用 atrm 而不用 at -d)
-v : 列出所有已经完成但尚未删除的指定
-c 将命令行上所列的作业送到标准输出。

at 5:30pm
at 17:30
at 17:30 today
at now + 5 hours
at now + 300 minutes
at 17:30 24.2.99
at 17:30 2/24/99
at 17:30 Feb 24
at 17:30 2/24/99

at now + 5 min
5 分钟后执行

at< /tmp/work 2:00 12/25/99
同-f 参数

$ at -f work 4pm + 3 days
在三天后下午 4 点执行文件 work 中的作业。

at 将显示任务执行的时间
检查邮件。确认工作完成了

336 ☆atq

atq [-V] [-q 队列] [-v]

337 ☆atrm

atrm [-V] 作业 [作业...]

338 ☆batch

batch 用低优先级运行作业，该命令几乎和 at 命令的功能完全相同，唯一的区别在于，at 命令是在指定时间，很精确的时刻执行指定命令;而 batch 却是在系统负载较低，资源比较空闲的时候执行命令。该命令适合于执行占用资源较多的命令。

batch [-V] [-q 队列] [-f 文件名] [-mv] [时间]

一般地说，不用为 batch 命令指定时间参数，因为 batch 本身的特点就是由系统决定执行任务的时间，如果用户再指定一个时间，就失去了本来的意义。

仍然使用<Ctrl+d>组合键来结束命令输入。而且 batch 和 at 命令都将自动转入后台，所以启动的时候也不需要加上&符号。

batch 等效于 at -q b
小于 80%CPU 才运行

339 ☆crontab

使用权限 ：所有使用者

使用方式 ：
crontab [-u user] filecrontab [-u user] { -l | -r | -e }

参数 ：
filecrontab 用于安装一个新的 crontab 文件，安装来源就是 filecrontab 所指的文件，如果使用“-”符号作为文件名，那就意味着使用标准输入作为安装来源。

-u 如果使用该选项，也就是指定了是哪个具体用户的 crontab 文件将被修改。如果不指定该选项，crontab 将默认是操作者本人的 crontab ，也就是执行该 crontab 命令的用户的 crontab 文件将被修改。但是请注意，如果使用了 su 命令再使用 crontab 命令很可能就会出现混乱的情况。所以如果是使用了 su 命令，最好使用-u 选项来指定究竟是哪个用户的 crontab 文件。
-l 在标准输出上显示当前的 crontab。

-r 删除当前的 crontab 文件。
-e 使用 VISUAL(比如说 setenv VISUAL joe)或者 EDITOR 环境变量所指的编辑器编辑当前的 crontab 文件。当结束编辑离开时，编辑后的文件将自动安装。

说明 ：
crontab 是用来让使用者在固定时间或固定间隔执行程序之用，换句话说，也就是类似使用者的时程表。-u user 是指设定指定 user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设定自己的时程表。

cron 命令会搜索/var/spool/cron 目录，文件以用户名命名的 crontab 文件，被找到的这种文件将载入内存。crontab -e 修改
cron 命令还将搜索/etc/crontab 文件,vi 直接修改
cron 启动以后，它将首先检查是否有用户设置了 crontab 文件，如果没有就转入“休眠”状态，释放系统资源。所以该后台进程占用资源极少。它每分钟“醒”过来一次，查看当前是否有需要运行的命令。命令执行结束后，任何输出都将作为邮件发送给 crontab 的所有者，或者是/etc/crontab 文件中 MAILTO 环境变量中指定的用户。

可以使用 crontab 命令的用户是有限制的。如果/etc/cron.allow 文件存在，那么只有其中列出的用户才能使用该命令(忽略 deny);如果该文件不存在但 cron.deny 文件存在，那么只有未列在该文件中的用户才能使用 crontab 命令;如果两个文件都不存在，那就取决于一些参数的设置，可能是只允许超级用户使用该命令，也可能是所有用户都可以使用该命令。

时程表的格式如下 ：
分钟 小时 日 月 周 [用户名] 命令

第一段应该定义的是：分钟，表示每个小时的第几分钟来执行。范围是从 0-59
第二段应该定义的是：小时，表示从第几个小时来执行，范围是从 0-23
第三段应该定义的是：日期，表示从每个月的第几天执行，范围从 1-31
第四段应该定义的是：月，表示每年的第几个月来执行，范围从 1-12
第五段应该定义的是：周，表示每周的第几天执行，范围从 0-6，其中 0 表示星期日。
每六段应该定义的是：用户名，也就是执行程序要通过哪个用户来执行，这个一般可以省略;
第七段应该定义的是：执行的命令和参数。

注：其中用户名可是省略，用户名定义的是程序用哪个用户来执行

如果有*代表的地方，表示全部，也就是说，每个月，每天，每星期都要执行。

让配置文件生效：如果让配置文件生效，还得重新启动 cron，切记，既然每个用户下的 cron 配置文件修改后。也要重新启动 cron 服务器。

/etc/init.d/crond restart

chkconfig --levels 35 crond on

同一时间，同时执行多个任务的定义方法
放在/etc 目录下的 cron.hourly 、cron.daily 、cron.weekly 、cron.monthly 目录中。凡是放进这些目录的可执行脚本，都能在约定的时间内准确执行。每个目录有每个目录的用途;

[root@localhost cron.daily]# touch httpd.sh
[root@localhost cron.daily]# chmod 755 httpd.sh
[root@localhost cron.daily]# echo "/etc/init.d/httpd restart" > httpd.sh

当 f1 为 * 时表示每分钟都要执行 program，f2 为 * 时表示每小时都要执行程式，其余类推
当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其余类推
当 f1 为 */n 时表示每 n 分钟个时间间隔执行一次，f2 为 */n 表示每 n 小时个时间间隔执行一次，其余类推
当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其余类推
a-b/x x 为步长,a 开始加上 x,直到 b

使用者也可以将所有的设定先存放在文件 filecrontab 中，用 crontab filecrontab 的方式来设定时程表。

每月每天每小时的第 0 分钟执行一次 /bin/ls :
0 7 * * * /bin/ls

在 12 月内, 每天的早上 6 点到 12 点中，每隔 20 分钟执行一次 /usr/bin/backup :
0 6-12/3 * 12 * /usr/bin/backup

周一到周五每天下午 5:00 寄一封信给 alex@domain.name :
0 17 * * 1-5 mail -s "hi" alex@domain.name < /tmp/maildata

每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"
20 0-23/2 * * * echo "haha"

注意 :

当程式在你所指定的时间执行后，系统会寄一封信给你，显示该程式执行的内容，若是你不希望收到这样的信，请在每一行空一格之后加上 > /dev/null 2>&1 即可。

编辑 test.cron
格式为 * * * * * cmd

crontab test.cron
安装这个文件
/var/spool/cron 下就有个新文件了

crontab -l #列出用户目前的 crontab。

340 ☆anacron

停电后补做
/etc/anacrontab

7 70 cron.weekly run-parts /etc/cron.weekly
7 天内没有做的话,启动 anacron 后 70 分钟运行

341 ☆ps

使用权限:所有使用者
使用方式:ps [options] [--help]
说明:显示瞬间行程 (process) 的动态

语法: ps [-aAcdefHjlmNVwy][acefghLnrsSTuvxX][-C <指令名称>][-g <群组名称>][-G <群组识别码>][-p <程序识别码>][p <程序识别码>][-s <阶段作业>][-t <终端机编号>][t <终端机编号>][-u <用户识别码>][-U <用户识别码>][U <用户名称>][-<程序识别码>][--cols <每列字符数>][--columns <每列字符数>][--cumulative][--deselect][--forest][--headers][--help][--info][--lines <显示列数>][--no-headers][--group <群组名称>][-Group <群组识别码>][--pid <程序识别码>][--rows <显示列数>][--sid <阶段作业>][--tty <终端机编号>][--user <用户名称>][--User <用户识别码>][--version][--width <每列字符数>]

补充说明：ps 是用来报告程序执行状况的指令，您可以搭配 kill 指令随时中断，删除不必要的程序。

参数:
-a 显示所有终端机下执行的程序，除了阶段作业领导者之外。
-A 显示所有程序。
-c 显示 CLS 和 PRI 栏位。
-C<指令名称> 指定执行指令的名称，并列出该指令的程序的状况。
-d 显示所有程序，但不包括阶段作业领导者的程序。
-e 此参数的效果和指定"A"参数相同。
-f 用 ASCII 字符显示树状结构，表达程序间的相互关系。
-g<群组名称> 此参数的效果和指定"-G"参数相同，当亦能使用阶段作业领导者的名称来指定。
-G<群组识别码> 列出属于该群组的程序的状况，也可使用群组名称来指定。

-H 显示树状结构，表示程序间的相互关系。
-j 或 j 采用作业格式显示程序状况。
-l 或 l 采用详细的格式来显示程序状况。
-m 或 m 显示所有的执行绪。
-n 以数字来表示 USER 和 WCHAN 栏位。
-N 显示所有的程序，除了执行 ps 指令终端机下的程序之外。
-p<程序识别码> 指定程序识别码，并列出该程序的状况。
-r 只列出现行终端机正在执行中的程序。
-s<阶段作业> 指定程序信号的格式，并列出隶属该阶段作业的程序的状况。
-S 列出程序时，包括已中断的子程序资料。
-t<终端机编号> 指定终端机编号，并列出属于该终端机的程序的状况。
-T 显示现行终端机下的所有程序。
-u<用户识别码> 此参数的效果和指定"-U"参数相同。
-U<用户识别码> 列出属于该用户的程序的状况，也可使用用户名称来指定。
-v 采用虚拟内存的格式显示程序状况。
-V 或 V 显示版本信息。
-w 或 w 采用宽阔的格式来显示程序状况。
-x 显示所有程序，不以终端机来区分。
-X 采用旧式的 Linux i386 登陆格式显示程序状况。
-y 配合参数"-l"使用时，不显示 F(flag)栏位，并以 RSS 栏位取代 ADDR 栏位。
-<程序识别码> 此参数的效果和指定"p"参数相同。
--cols<每列字符数> 设置每列的最大字符数。
--columns<每列字符数> 此参数的效果和指定"--cols"参数相同。
--cumulative 此参数的效果和指定"S"参数相同。
--deselect 此参数的效果和指定"-N"参数相同。
--forest 此参数的效果和指定"f"参数相同。
--headers 重复显示标题列。
--help 在线帮助。
--info 显示排错信息。
--lines<显示列数> 设置显示画面的列数。
--no-headers 此参数的效果和指定"h"参数相同，只在列表格式方面稍有差异。
--group<群组名称> 此参数的效果和指定"-G"参数相同。
--Group<群组识别码> 此参数的效果和指定"-G"参数相同。
--pid<程序识别码> 此参数的效果和指定"-p"参数相同。
--rows<显示列数> 此参数的效果和指定"--lines"参数相同。
--sid<阶段作业> 此参数的效果和指定"-s"参数相同。
--tty<终端机编号> 此参数的效果和指定"-t"参数相同。
--user<用户名称> 此参数的效果和指定"-U"参数相同。
--User<用户识别码> 此参数的效果和指定"-U"参数相同。
--version 此参数的效果和指定"-V"参数相同。
--widty<每列字符数> 此参数的效果和指定"-cols"参数相同。

--sort X[+|-] key [，[+|-] key [，…]] 从 SORT KEYS 段中选一个多字母键。"+"字符是可选的，因为默认的方向就是按数字升序或者词典顺序。比如: ps -jax -sort=uid，-ppid，+pid。
-O[+|-] k1 [，[+|-] k2 [，…]] 根据 SHORT KEYS、k1、k2 中快捷键指定的多级排序顺序显示进程列表。对于 ps 的不同格式都存在着默认的顺序指定。这些默认顺序可以被用户的指定所覆盖。其中"+"字符是可选的，"-"字符是倒转指定键的方向。

短格式
长格式
说 明

c
cmd
可执行的简单名称

C
cmdline
完整命令行

f
flags
长模式标志

g
pgrp
进程的组 ID

G
tpgid
控制 tty 进程组 ID

j
cutime
累计用户时间

J
cstime
累计系统时间

k
utime
用户时间

K
stime
系统时间

m
min_flt
次要页错误的数量

M
maj_flt
主要页错误的数量

n
cmin_flt
累计次要页错误

N
cmaj_flt
累计主要页错误

o
session
对话 ID

p
pid
进程 ID

P
ppid
父进程 ID

r
rss
驻留大小

R
resident
驻留页

s
size

内存大小(千字节)

S
share
共享页的数量

t
tty
tty 次要设备号

T
start_time
进程启动的时间

U
uid
UID

u
user
用户名

v
vsize
总的虚拟内存数量(字节)

y
priority
内核调度优先级

au(x) 输出格式 :

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
USER: 行程拥有者
PID: pid
PPID:父进程;
%CPU: 占用的 CPU 使用率
%MEM: 占用的内存使用率
NI:进程的 NICE 值，数值大，表示较少占用 CPU 时间;
VSZ: 占用的虚拟内存大小
RSS: 占用的内存大小
TTY: 终端的次要装置号码 (minor device number of tty)

STAT: 该行程的状态:
D: 不可中断的休眠 (通常进行 I/O 动作)
R: 正在执行中
S: 静止状态
T: 暂停执行
Z: 不存在但暂时无法消除,僵尸进程
W: 没有足够的内存分页可分配,进入内存交换
X:死掉的进程（从来没见过）
<: 高优先序的行程
N: 低优先序的行程
L: 有内存分页分配并锁在内存内 (即时系统或捱 A I/O)
WCHAN:正在等待的进程资源;
START: 进程开始时间
TIME: 执行的时间
COMMAND:所执行的指令

我们常用的选项是组合是 aux 或 lax，还有参数 f 的应用;

ps aux 最常用
ps auxf 父进和子进程的关系友好判断

342 ☆pgrep

pgrep 是通过程序的名字来查询进程的工具，一般是用来判断程序是否正在运行。在服务器的配置和管理中，这个工具常被应用，简单明了;

用法:
#pgrep 参数选项 程序名

常用参数
-l 列出程序名和进程 ID;
-o 进程起始的 ID;
-n 进程终止的 ID;

343 ☆kill

使用权限:所有使用者

使用方式:
kill [-s signal | -p] [-a] pid ...
kill -l [signal]

说明:kill 送出一个特定的信号 (signal) 给行程 id 为 pid 的行程根据该信号而做特定的动作, 若没有指定, 预设是送出终止 (TERM) 的信号

补充说明：kill 可将指定的信息送至程序。预设的信息为 SIGTERM(15)，可将指定程序终止。若仍无法终止该程序，可使用 SIGKILL(9)信息尝试强制删除程序。程序或工作的编号可利用 ps 指令或 jobs 指令查看。

参数：
-l <信息编号> :若不加<信息编号>选项，则-l 参数会列出全部的信息名称。显示信号名称列表，这也可以在/usr/include/linux/signal.h 文件中找到。
-s <信息名称或编号> :指定要送出的信息。
-p :指定 kill 命令只是显示进程的 pid，并不真正送出结束信号。
[程序]可以是程序的 PID 或是 PGID，也可以是工作编号。

$ kill -l
看清单
$ kill -19 %1
暂停
$ kill -18 %1
继续

使用 kill 命令的 SIGTERM（15）信号，也是 kill 的默认信号
kill %2 %3

结束或终止后台中的进程
语法:kill %n

344 ☆killall

killall 通过程序的名字，直接杀死所有进程

用法：killall 正在运行的程序名

killall 也和 ps 或 pgrep 结合使用，比较方便;通过 ps 或 pgrep 来查看哪些程序在运行;

[root@panda ~]# pgrep -l gaim
2979 gaim
[root@panda ~]# killall gaim

345 ☆pkill

pkill 和 killall 应用方法差不多，也是直接杀死运行中的程序;如果您想杀掉单个进程，请用 kill 来杀掉。

#pkill 正在运行的程序名

[root@panda ~]# pgrep -l gaim
2979 gaim
[root@panda ~]# pkill gaim

346 ☆xkill

xkill 是在桌面用的杀死图形界面的程序。
当 xkill 运行时出来和个人脑骨的图标，哪个图形程序崩溃一点就 OK 了。如果您想终止 xkill ，就按右键取消;

xkill 调用方法：
[root@localhost ~]# xkill

347 ☆skill

使用权限:所有使用者

使用方式: skill [signal to send] [options] 选择程序的规则

说明:

送个讯号给正在执行的程序,预设的讯息为 TERM (中断) , 较常使用的讯息为 HUP , INT , KILL , STOP , CONT ,和 0

讯息有三种写法:分别为 -9 , -SIGKILL , -KILL , 可以使用 -l 或 -L 已列出可使用的讯息。

一般参数:
-f 快速模式/尚未完成
-i 互动模式/ 每个动作将要被确认
-v 详细输出/ 列出所选择程序的资讯
-w 智能警告讯息/ 尚未完成
-n 没有动作/ 显示程序代号

参数:选择程序的规则可以是, 终端机代号,使用者名称,程序代号,命令名称。
-t 终端机代号 (tty 或 pty)

-u 使用者名称
-p 程序代号 (pid)
-c 命令名称 可使用的讯号:

以下列出已知的讯号名称,讯号代号,功能。

名称 (代号) 功能/ 描述
ALRM 14 离开
HUP 1 离开
INT 2 离开
KILL 9 离开/ 强迫关闭
PIPE 13 离开
POLL 离开
PROF 离开
TERM 15 离开
USR1 离开
USR2 离开
VTALRM 离开
STKFLT 离开/ 只适用于 i386, m68k, arm 和 ppc 硬体
UNUSED 离开/ 只适用于 i386, m68k, arm 和 ppc 硬体
TSTP 停止 /产生与内容相关的行为
TTIN 停止 /产生与内容相关的行为
TTOU 停止 /产生与内容相关的行为
STOP 停止 /强迫关闭
CONT 从新启动 /如果在停止状态则从新启动,否则忽略
PWR 忽略 /在某些系统中会离开
WINCH 忽略
CHLD 忽略
ABRT 6 核心
FPE 8 核心
ILL 4 核心
QUIT 3 核心
SEGV 11 核心
TRAP 5 核心
SYS 核心 /或许尚未实作
EMT 核心 /或许尚未实作
BUS 核心 /核心失败
XCPU 核心 /核心失败
XFSZ 核心 /核心失败

范例:

停止所有在 PTY 装置上的程序
skill -KILL -v pts/*

停止三个使用者 user1 , user2 , user3
skill -STOP user1 user2 user3

348 ☆top

使用权限:所有使用者

使用方式:top [-] [d delay] [q] [c] [S] [s] [i] [n] [b]

说明:即时显示 process 的动态
补充说明：执行 top 指令可显示目前正在系统中执行的程序，并通过它所提供的互动式界面，用热键加以管理。
top 命令和 ps 命令的基本作用是相同的，显示系统当前的进程和其他状况;但是 top 是一个动态显示过程，即可以通过用户按键来不断刷新当前状态。如果在前台执行该命令，它将独占前台，直到用户终止该程序为止。

参数:
d<间隔秒数> :显示两次刷新时间的间隔,单位以秒计算。当然用户可以使用 s 交互命令来改变之。
q : 没有任何延迟的显示速度，如果使用者是有 superuser 的权限，则 top 将会以最高的优先序执行
c : 显示命令行，而不仅仅是命令名,切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完整的路径与名称 S : 累积模式，会将己完成或消失的子行程 (dead child process) 的 CPU time 累积起来
s : 安全模式，将交谈式指令取消, 避免潜在的危机
S 使用累计模式，输出每个进程的总的 CPU 时间，包括已死的子进程;其效果类似 ps 指令的"-S" 参数。
i : 不显示任何闲置 (idle) 或无用 (zombie) 的行程
n<执行次数> :显示更新次数，然后退出
n : 更新的次数，完成后将会退出 top
b : 批次档模式，但不能接受命令行输入,搭配 "n" 参数一起使用，可以用来将 top 的结果输出到文件内 p PID 仅监视指定进程的 ID;PID 是一个数值;

交互式命令键位：

space 立即更新;
Ctrl+L 擦除并且重写屏幕。
h,? 显示有关安全模式及累积模式的帮助信息;
c 切换到命令名显示，或显示整个命令（包括参数）;
f,F 增加显示字段，或删除显示字段;

k 提示输入要杀死的进程 ID，目的是用来杀死该进程（默人信号为 15）
i 禁止空闲进程和僵尸进程;
l 切换到显示负载平均值和正常运行的时间等信息;
m 切换到内存信息，并以内存占用大小排序;
n 提示显示的进程数，比如输入 3，就在整屏上显示 3 个进程;
o,O 改变显示字段的顺序;
r 把 renice 应用到一个进程，提示输入 PID 和 renice 的值;默认值是 10。
s 改变两次刷新时间间隔，以秒为单位;输入 0 值则系统将不断刷新，默认值是 5s.
t 切换到显示进程和 CPU 状态的信息;
A 按进程生命大小进行排序，最新进程显示在最前;
M 按内存占用大小排序，由大到小;
N 以进程 ID 大小排序，由大到小;
P 按 CPU 占用情况排序，由大到小
S 切换到累积时间模式;
T 按时间／累积时间对任务排序;
W 把当前的配置写到~/.toprc 中;
q 退出程序。

top 命令显示的项目很多，默认值是每 5 秒更新一次，当然这是可以设置的。显示的各项目为:

uptime 该项显示的是系统启动时间、已经运行的时间和三个平均负载值(最近 1 秒，5 秒，15 秒的负载值)。
processes 自最近一次刷新以来的运行进程总数。当然这些进程被分为正在运行的，休眠的，停止的等很多种类。进程和状态显示可以通过交互命令 t 来实现。
CPU states 显示用户模式，系统模式，优先级进程(只有优先级为负的列入考虑)和闲置等各种情况所占用 CPU 时间的百分比。优先级进程所消耗的时间也被列入到用户和系统的时间中，所以总的百分比将大于 100%。
Mem 内存使用情况统计，其中包括总的可用内存，空闲内存，已用内存，共享内存和缓存所占内存的情况。
Swap 交换空间统计，其中包括总的交换空间，可用交换空间，已用交换空间。
PID 每个进程的 ID。
PPID 每个进程的父进程 ID。
UID 每个进程所有者的 UID 。
USER 每个进程所有者的用户名。
PRI 每个进程的优先级别。
NI 该进程的优先级值。
SIZE 该进程的代码大小加上数据大小再加上堆栈空间大小的总数。单位是 KB。
TSIZE 该进程的代码大小。对于内核进程这是一个很奇怪的值。
DSIZE 数据和堆栈的大小。
TRS 文本驻留大小。
D 被标记为"不干净"的页项目。
LIB 使用的库页的大小。对于 ELF 进程没有作用。

RSS 该进程占用的物理内存的总数量，单位是 KB。
SHARE 该进程使用共享内存的数量。
STAT 该进程的状态。其中 S 代表休眠状态;D 代表不可中断的休眠状态;R 代表运行状态;Z 代表僵死状态;T 代表停止或跟踪状态。
TIME 该进程自启动以来所占用的总 CPU 时间。如果进入的是累计模式，那么该时间还包括这个进程子进程所占用的时间。且标题会变成 CTIME。
%CPU 该进程自最近一次刷新以来所占用的 CPU 时间和总时间的百分比。
%MEM 该进程占用的物理内存占总内存的百分比。
COMMAND 该进程的命令名称，如果一行显示不下，则会进行截取。内存中的进程会有一个完整的命令行。

范例:
显示更新十次后退出 ；
top -n 10

使用者将不能利用交谈式指令来对行程下命令 ：
top -s

将更新显示二次的结果输入到名称为 top.log 的文件里 ：
top -n 2 -b > top.log

349 ☆nice

使用权限:所有使用者

使用方式:
nice [-n adjustment] [-adjustment] [--adjustment=adjustment] [--help] [--version] [command [arg...]]
nice [-n <优先等级>][--help][--version][执行指令]

说明:以更改过的优先序来执行程式, 如果未指定程式, 则会印出目前的排程优先序, 内定的 谦让度 adjustment 为 10, 范围为 -20 (最高优先序) 到 19 (最低优先序)

补充说明：nice 指令可以改变程序执行的优先权等级。

参数： -n<优先等级>或-<优先等级>或--adjustment=<优先等级> 设置欲执行的指令的优先权等级。等级的范围从-20-19，其中-20 最高，19 最低，只有系统管理者可以设置负数的等级。
--help 在线帮助。
--version 显示版本信息。

将 ls 的优先序为 1 并执行 ：
nice -n 1 ls

注意 ： 优先序 (priority) 为作业系统用来决定 CPU 分配的参数，Linux 使用『回合制 (round-robin)』的演算法来做 CPU 排程，优先序越高，所可能获得的 CPU 时间就越多。

350 ☆renice

名称:renice
使用权限:所有使用者

使用方式:renice priority [[-p] pid ...] [[-g] pgrp ...] [[-u] user ...]
功能说明：调整优先权。
说明:重新指定一个或多个行程(Process)的优先序(一个或多个将根据所下的参数而定)
语法：renice [优先等级][-g <程序群组名称>...][-p <程序识别码>...][-u <用户名称>...]

补充说明：renice 指令可重新调整程序执行的优先权等级。预设是以程序识别码指定程序调整其优先权，您亦可以指定程序群组或用户名称调整优先权等级，并修改所有隶属于该程序群组或用户的程序的优先权。等级范围从-20--19，只有系统管理者可以改变其他用户程序的优先权，也仅有系统管理者可以设置负数等级。

参数：
-g <程序群组名称> 使用程序群组名称，修改所有隶属于该程序群组的程序的优先权。
-p <程序识别码> 改变该程序的优先权等级，此参数为预设值。
-u <用户名称> 指定用户名称，修改所有隶属于该用户的程序的优先权。

注意 ： 每一个行程(Process)都有一个唯一的 (unique) id

$ renice -number PID

其中，参数 number 与 nice 命令的 number 意义相同。

注:
(1)用户只能对自己所有的进程使用 renice 命令。
(2)root 用户可以在任何进程上使用 renice 命令。
(3)只有 root 用户才能提高进程的优先权。

范例:
将行程 id 为 987 及 32 的行程与行程拥有者为 daemon 及 root 的优先序号码设为 1 :
renice +1 987 -u daemon root -p 32

351 ☆pstree

(process status tree)

使用权限:所有使用者
使用方式:
pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-G|-U] [pid|user]
pstree -V
说明:将所有行程以树状图显示, 树状图将会以 pid (如果有指定) 或是以 init 这个基本行程为根 (root) ,如果有指定使用者 id , 则树状图会只显示该使用者所拥有的行程
参数:
-a 显示该行程的完整指令及参数, 如果是被内存置换出去的行程则会加上括号
-c 如果有重覆的行程名, 则分开列出 (预设值是会在前面加上 *
功能说明：以树状图显示程序。

语法：pstree [-acGhlnpuUV][-H <程序识别码>][<程序识别码>/<用户名称>]

补充说明：pstree 指令用 ASCII 字符显示树状结构，清楚地表达程序间的相互关系。如果不指定程序识别码或用户名称，则会把系统启动时的第一个程序视为基层，并显示之后的所有程序。若指定用户名称，便会以隶属该用户的第一个程序当作基层，然后显示该用户的所有程序。

参数：
-a 显示每个程序的完整指令，包含路径，参数或是常驻服务的标示。
-c 不使用精简标示法。
-G 使用 VT100 终端机的列绘图字符。
-h 列出树状图时，特别标明现在执行的程序。
-H<程序识别码> 此参数的效果和指定"-h"参数类似，但特别标明指定的程序。
-l 采用长列格式显示树状图。
-n 用程序识别码排序。预设是以程序名称来排序。
-p 显示程序识别码。
-u 显示用户名称。
-U 使用 UTF-8 列绘图字符。
-V 显示版本信息。

范例:

pstree

init-+-amd
|-apmd
|-atd
|-httpd---10*[httpd]
%pstree -p
init(1)-+-amd(447)
|-apmd(105)
|-atd(339)

%pstree -c
init-+-amd
|-apmd
|-atd
|-httpd-+-httpd
| |-httpd
| |-httpd
| |-httpd
....

352 ☆gitps

(gnu interactive tools process status)

功能说明：报告程序状况。

语法：gitps [acefgjlnrsSTuvwxX][p <程序识别码>][t <终端机编号>][U <帐号名称>]

补充说明：gitps 是用来报告并管理程序执行的指令，基本上它就是通过 ps 指令来报告，管理程序，也能通过 gitps 指令随时中断，删除不必要的程序。因为 gitps 指令会去执行 ps 指令，所以其参数和 ps 指令相当类似。

参数：
a 显示现行终端机下的所有程序，包括其他用户的程序。
c 列出程序时，显示每个程序真正的指令名称，而不包含路径，参数或是常驻服务的标示.
e 列出程序时，显示每个程序所使用的环境变量。
f 用 ASCII 字符显示树状结构，表达程序间的相互关系。
g 显示现行终端机下的所有程序，包括群组领导者的程序。
j 采用工作控制的格式来显示程序状况。
l 采用纤细的格式来显示程序状况。
n 以数字来表示 USER 和 WCHAN 栏位。
p<程序识别码> 指定程序识别码，并列出该程序的状况。
r 只列出现行终端机正在执行中的程序。
s 采用程序信号的格式显示程序状况。
S 列出程序时，包括已中断的子程序信息。
t<终端机机标号> 指定终端机编号，并列出属于该终端机的程序的状况。
T 显示现行终端机下的所有程序。
u 以用户为主的格式来显示程序状况。
U<帐号名称> 列出属于该用户的程序的状况。
v 采用虚拟内存的格式显示程序状况。
w 采用宽阔的格式来显示程序状况。
x 显示所有程序，不以终端机来区分。

X 采用旧试的 Linux i386 登陆格式显示程序状况。

353 ☆tload

功能说明：显示系统负载状况。

语法：tload [-V][-d <间隔秒数>][-s <刻度大小>][终端机编号]

补充说明：tload 指令使用 ASCII 字符简单地以文字模式显示系统负载状态。假设不给予终端机编号，则会在执行 tload 指令的终端机显示负载情形。

参数：
-d<间隔秒数> 设置 tload 检测系统负载的间隔时间，单位以秒计算。
-s<刻度大小> 设置图表的垂直刻度大小，单位以列计算。
-V 显示版本信息。

354 ☆swatch

(simple watcher)

功能说明：系统监控程序。

语法：swatch [-A <分隔字符>][-c <设置文件>][-f <记录文件>][-I <分隔字符>][-P <分隔字符>][-r <时间>][-t <记录文件>]

补充说明：swatch 可用来监控系统记录文件，并在发现特定的事件时，执行指定的动作。swatch 所监控的事件以及对应事件的动作都存放在 swatch 的配置文件中。预设的配置文件为拥护根目录下的.swatchrc。然而在 Red Hat Linux 的预设用户根目录下并没有.swatchrc 配置文件，您可将 /usr/doc/swatch-2.2/config_files/swatchrc.personal 文件复制到用户根目录下的.swatchrc，然后修改.swatchrc 所要监控的事件及执行的动作。

参数：
-A<分隔字符> 预设配置文件中，动作的分隔字符，预设为逗号。
-c 设置文件> 指定配置文件，而不使用预设的配置文件。
-f 记录文件> 检查指定的记录文件，检查完毕后不会继续监控该记录文件。
-I 分隔字符> 指定输入记录的分隔字符，预设为换行字符。
-P 分隔字符> 指定配置文件中，事件的分隔字符，预设为逗号。
-r 时间> 在指定的时间重新启动。
-t<记录文件> 检查指定的记录文件，并且会监控加入记录文件中的后继记录。

355 ☆suspend

功能说明：暂停执行 shell。

语法：suspend [-f]

补充说明：suspend 为 shell 内建指令，可暂停目前正在执行的 shell。若要恢复，则必须使用 SIGCONT 信息。

参数：
-f 若目前执行的 shell 为登入的 shell，则 suspend 预设无法暂停此 shell。若要强迫暂停登入的 shell，则必须使用-f 参数。

356 ☆sysstat

Sysstat 是一个软件包，包含监测系统性能及效率的一组工具，这些工具对于我们收集系统性能数据，比如 CPU 使用率、硬盘和网络吞吐数据，这些数据的收集和分析，有利于我们判断系统是否正常运行，是提高系统运行效率、安全运行服务器的得力助手;

Sysstat 软件包集成如下工具：
* iostat 工具提供 CPU 使用率及硬盘吞吐效率的数据;
* mpstat 工具提供单个处理器或多个处理器相关数据;
* sar 工具负责收集、报告并存储系统活跃的信息;
* sa1 工具负责收集并存储每天系统动态信息到一个二进制的文件中。它是通过计划任务工具 cron 来运行，是为 sadc 所设计的程序前端程序;
* sa2 工具负责把每天的系统活跃性息写入总结性的报告中。它是为 sar 所设计的前端 ，要通过 cron 来调用
* sadc 是系统动态数据收集工具，收集的数据被写一个二进制的文件中，它被用作 sar 工具的后端;
* sadf 显示被 sar 通过多种格式收集的数据;

安装
[root@localhost ~]#rpm -ivh sysstat*.rpm

关于 Sysstat 计划任务;
如果您想得到 Sysstat 工具集所收集的系统信息自动存为某个文件中，你必须通过 cron 为 sa1 和 sa2 做计划任务。我们可以通过修改用户的 crontab。在默认的情况下，Sysstat 历史信息将被存放在/var/log/sa 文件中。

在 root 用户，通过 crontab -e 来添加下面的一段;
8am-7pm activity reports every 10 minutes during weekdays
0 8-18 * * 1-5 /usr/lib/sa/sa1 600 6 &

```
# 7pm-8am activity reports every hour during weekdays
0 19-7 * * 1-5 /usr/lib/sa/sa1 &
# Activity reports every hour on Saturday and Sunday
0 * * * 0,6 /usr/lib/sa/sa1 &
# Daily summary prepared at 19:05 5 19 * * * /usr/lib/sa/sa2 -A &
```

创建 Sysstat 的启动脚本;

```
[root@localhost ~]# touch /etc/rc.d/init.d/sysstat
[root@localhost ~]# vi /etc/rc.d/init.d/sysstat
```

```
#!/bin/sh
# Begin $rc_base/init.d/sysstat
# Based on sysklogd script from LFS-3.1 and earlier.
# Rewritten by Gerard Beekmans - gerard@linuxfromscratch.org
  . /etc/sysconfig/rc
  . $rc_functions
case "$1" in
        start)
                echo "Calling the system activity data collector (sadc)..."
                /usr/lib/sa/sadc -F -L -
                evaluate_retval
                ;;
        *)
                echo "Usage: $0 start"
                exit 1
                ;;
esac
# End $rc_base/init.d/sysstat
```

```
[root@localhost ~]# chmod 755 /etc/rc.d/init.d/sysstat
[root@localhost ~]# ln -sf /etc/rc.d/init.d/sysstat /etc/init.d/sysstat
```

有了 Sysstat 的守护进程，这样我们开机后，Sysstat 的守护进程，就时时刻刻的为我们服务了。sa 、sa1 或 sa2 自动把信息存在 /var/log/sa 目录的二进制文件中，我们可以通过 sar 工具来提取这些系统信息的历史;

当然我们也可以通过手动的方法来打开 Sysstat 的守护程序，也就是我们前面所制作的 sysstat;

```
[root@localhost ~]# /etc/rc.d/init.d/sysstat start
```

下面的方法也行;

```
[root@localhost ~]# /usr/lib/sa/sa1
[root@localhost ~]# /usr/lib/sa/sa2
```

Sysstat 工具集介绍

sadc 工具
sadc 位于 /usr/lib/sa 目录中，如果你没有设置可执行路径，要用绝对路径来运行比较方便 ，/usr/lib/sa/sadc;
sadc 是把数据写在一个二进制的文件中，如果想查看数据内容，需要用 sadf 工具来显示;

sadc 的用法

/usr/lib/sa/sadc [-d] [-F] [-I] [-L] [-V] [interval [count]] [outfile]

参数说明：
-d 报告硬盘设置的相关统计;
-F 强制把数据写入文件;
-I 报告所有系统中断数据;
interval 表示时间间隔，单位是秒，比如 3 ;
count 统计数据的次数，也是一个数字;
outfile 输出统计到 outfile 文件;

注意：此工具中的参数都是可选的，如果没有指定任何参数，比如 /usr/lib/sa/sadc - ，则会输出数据到 /var/log/sa/ 目录下的一个文件中。我们要通过 sadf 或 sar 工具来查看;

[root@panda ~]# /usr/lib/sa/sadc -
[root@panda ~]# ls /var/log/sa 注：列出所有 sa 目录下的文件，根据文件的时间来判断哪个文件是最新的;
[root@panda ~]# sar -f /var/log/sa/sa12
或
[root@panda ~]# sadf /var/log/sa/sa12

举例：我们想把 sadc 收集到的数据写到一个指定的文件中;

[root@localhost ~]# /usr/lib/sa/sadc 1 10 sa000
[root@localhost ~]# sar -f sa000

Linux 2.6.15-1.2054_FC5 (localhost.localdomain) 2006 年 05 月 12 日

09 时 15 分 30 秒	CPU	%user	%nice	%system	%iowait	%idle
09 时 15 分 31 秒	all	3.00	0.00	0.00	1.00	96.00
09 时 15 分 32 秒	all	0.00	0.00	0.00	0.00	100.00
09 时 15 分 33 秒	all	0.00	0.00	0.00	0.00	100.00
09 时 15 分 34 秒	all	0.00	0.00	0.00	0.00	100.00

09 时 15 分 35 秒	all	0.00	0.00	0.00	0.00	100.00
09 时 15 分 36 秒	all	0.00	0.00	0.00	0.00	100.00
09 时 15 分 37 秒	all	0.00	0.00	0.00	0.00	100.00
09 时 15 分 38 秒	all	0.00	0.00	0.00	0.00	100.00
09 时 15 分 39 秒	all	0.00	0.00	0.00	0.00	100.00
Average:	all	0.33	0.00	0.00	0.11	99.56

注解：我们用 sadc 收集系统动态数据，让它收集 1 秒之内的 10 次动态信息; 然后通过 sar 工具来查看系统的状态。也可以用 sadf 来查看所收集的数据，但不是太直观。您自己尝试一下看看。查看 sa000 文件，用 sadf sa000 ;

sar 工具;
sar 工具比较强大，既能收集系统 CPU、硬盘、动态数据，也能显示动态显示，更能查看二进制数据文件;sar 的应用比较多，而且也比较复杂，数据更为精确。我们只了解一下常用的内容就行，大多数内容我们了解就行;

用法：
sar [参数选项]

参数说明：

-A 显示所有历史数据，通过读取/var/log/sar 目录下的所有文件，并把它们分门别类的显示出来;
-b 通过设备的 I/O 中断读取设置的吞吐率;
-B 报告内存或虚拟内存交换统计;
-c 报告每秒创建的进程数;
-d 报告物理块设备（存储设备）的写入、读取之类的信息，如果直观一点，可以和 p 参数共同使用，-dp
-f 从一个二进制的数据文件中读取内容，比如 sar -f filename
-i interval 指定数据收集的时间，时间单位是秒;
-n 分析网络设备状态的统计，后面可以接的参数有 DEV、EDEV、NFS、NFSD、SOCK 等。比如-n DEV
-o 把统计信息写入一个文件，比如 -o filename ;
-P 报告每个处理器应用统计，用于多处理器机器，并且启用 SMP 内核才有效;
-p 显示友好设备名字，以方便查看，也可以和-d 和-n 参数结合使用，比如 -dp 或-np
-r 内存和交换区占用统计;
-R
-t 这个选项对从文件读取数据有用，如果没有这个参数，会以本地时间为标准 读出;
-u 报告 CPU 利用率的参数;
-v 报告 inode, 文件或其它内核表的资源占用信息;
-w 报告系统交换活动的信息; 每少交换数据的个数;
-W 报告系统交换活动吞吐信息;
-x 用于监视进程的，在其后要指定进程的 PID 值;

-X 用于监视进程的，但指定的应该是一个子进程 ID;

sar 应用举例;

实例一： 如果只用 sar 命令，sar 就是读取 /var/log/sa 目录下最近系统状态文件。
[root@localhost ~]# sar
[root@localhost ~]# sar -A 注：读取/var/log/sa 目录下所有文件数据;

如果我们想知道 CPU 的利用率;动态更新;下面的例子是每秒更新一次数据，总共更新五次;
[root@localhost ~]# sar -u 1 5
Linux 2.6.15-1.2054_FC5 (localhost.localdomain) 2006 年 05 月 12 日

时间	CPU	利用率	nice 值	系统占用	IO 占用	空闲
11 时 19 分 34 秒	CPU	%user	%nice	%system	%iowait	%idle
11 时 19 分 35 秒	all	2.97	0.00	0.00	0.00	97.03
11 时 19 分 36 秒	all	11.11	0.00	9.09	0.00	79.80
11 时 19 分 37 秒	all	21.78	0.00	6.93	0.00	71.29
11 时 19 分 38 秒	all	15.00	0.00	0.00	0.00	85.00
11 时 19 分 39 秒	all	8.00	0.00	0.00	0.00	92.00
Average:	all	11.78	0.00	3.19	0.00	85.03

注解：

CPU：表示机器内所有的 CPU;
%user 表示 CPU 的利用率;
%nice 表示 CPU 在用户层优先级的百分比，0 表示正常;
%system 表示当系统运行时，在用户应用层上所占用的 CPU 百分比;
%iowait 表示请求硬盘 I/0 数据流出时，所占用 CPU 的百分比;
%idle 表示空闲 CPU 百分比，值越大系统负载越低;

您可以 CPU 利用率的动态信息输出到一个文本文件中，然后通过 more 来查看。
[root@localhost ~]# sar -u 1 5 > sar000.txt
[root@localhost ~]# more sar000.txt

也可以输出到一个二进制的文件中，然后通过 sar 来查看;
[root@localhost ~]# sar -u 1 5 -o sar002
[root@localhost ~]# sar -f sar002

注：如果您把数据通过-o filename 输出到一个二进制的文件中，是不能用文件内容查看工具 more 、less 或 cat 来查看的，应该用 sar 工具来查看，要加-f 参数;

实例二：查看网络设备的吞吐情况;

比如我们让数据每秒更新一次，总共更新十次;
[root@localhost ~]# sar -n DEV 2 5
时间 IFACE rxpck/s txpck/s rxbyt/s txbyt/s rxcmp/s txcmp/s rxmcst/s

第一字段：时间;
IFACE：设备名;
rxpck/s:每秒收到的包;
rxbyt/s：每秒收到的所有包的体积;
txbyt/s：每秒传输的所有包的体积;
rxcmp/s：每秒收到数据切割压缩的包总数;
txcmp/s :每秒传输的数据切割压缩的包的总数;
rxmcst/s: 每秒收到的多点传送的包;

如果我们从事提取 eth0 设备（也就是网卡 eth0)的信息;我们应该用 grep 来过滤。然后再显示出来;
[root@localhost ~]# sar -n DEV 2 5 |grep eth0
11 时 52 分 37 秒 eth0 1.00 1.00 97.51 97.51 0.00 0.00 0.00
11 时 52 分 39 秒 eth0 1.01 1.01 98.49 98.49 0.00 0.00 0.00
11 时 52 分 41 秒 eth0 1.00 1.00 98.00 98.00 0.00 0.00 0.00
11 时 52 分 43 秒 eth0 1.00 1.00 98.00 98.00 0.00 0.00 0.00
11 时 52 分 45 秒 eth0 1.00 1.00 98.00 98.00 0.00 0.00 0.00
Average： eth0 1.00 1.00 98.00 98.00 0.00 0.00 0.00

如果想知道网络设备错误报告，也就就是用来查看设备故障的。应该用 EDEV;比如下面的例子;
[root@localhost ~]# sar -n EDEV 2 5

iostat
iostat 是用来显示 系统即时系统，比如 CPU 使用率，硬盘设备的吞吐率;

[root@localhost ~]# iostat
Linux 2.6.15-1.2054_FC5 (localhost.localdomain) 2006 年 05 月 12 日

avg-cpu: %user %nice %system %iowait %idle
7.24 0.00 0.99 0.35 91.43

Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
hda 1.46 28.43 21.43 710589 535680

mpstat

mpstat 提供多处理器系统中的 CPU 的利用率的统计;mpstat 也可以加参数，用-P 来指定哪个 CPU，处理器的 ID 是从 0 开始的。下面的例子是查看两个处理器，每二秒数据更新一次，总共要显示 10 次数据;

[root@localhost ~]# mpstat -P 0 2 10 注：查看第一个 CPU
[root@localhost ~]# mpstat -p 1 2 10 注：查看第二个 CPU
</code >
<code>
[root@localhost ~]# mpstat 2 10 注：查看所有 CPU;

sdaf
sdaf 能从二进制文件中提取 sar 所收集的数据;这个大家知道就行了。显示的并不是友好的格式;
[root@localhost ~]# sar -u 2 5 -o sar003
[root@localhost ~]# sadf sar003

相对来说，用 sar 来读取输出文件的内容更好;比如下面的;
[root@localhost ~]# sar -f sar003

vmstat 即时显示内存工具;
vmstat 是一个即时显示内存使用情况的工具;

vmstat 使用方法：

vmstat [-V] [-n] [delay [count]]
-V 显示 vmstat 的版本;
-n causes the headers not to be reprinted regularly.
-a 显示所有激活和未激活内存的状态;print inactive/active page stats.
-d 显示硬盘统计信息;prints disk statistics
-D 显示硬盘分区表;prints disk table
-p 显示硬盘分区读写状态等;prints disk partition statistics
-s 显示内存使用情况;prints vm table
-m prints slabinfo
-S 定义单位，k K
delay 是两次刷新时间间隔;
单位体积： k:1000 K:1024 m:1000000 M:1048576 (默认是 K)
count 刷新次数;

357 ☆expr

名称:expr

使用权限:所有使用者
字串长度

shell>> expr length "this is a test"
14

数字商数

shell>> expr 14 % 9
5

从位置处抓取字串

shell>> expr substr "this is a test" 3 5
is is

数字串 only the first character

shell>> expr index "testforthegame" e
2

字串真实重现

shell>> expr quote thisisatestformela
thisisatestformela

358 ☆clear

clear 命令的功能是清除屏幕上的信息，它类似于 DOS 中的 cls 命令。清屏后，提示符移动到屏幕左上角。

$ clear

359 ☆cal

使用权限:所有使用者

使用方式:cal [-mjy] [month [year]]

说明:

显示日历。若只有一个参数，则代表年份(1-9999)，显示该年的年历。年份必须全部写出:``cal 89\ 将不会是显示 1989 年的年历。使用两个参数，则表示月份及年份。若没有参数则显示这个月的月历。
1752 年 9 月第 3 日起改用西洋新历，因这时大部份的国家都采用新历，有 10 天被去除，所以该月份的月历有些不同。在此之前为西洋旧历。

参数:

-m : 以星期一为每周的第一天方式显示。
-j : 以凯撒历显示，显示出给定月中的每一天是一年中的第几天（从 1 月 1 日算起）。
-y : 显示今年年历。

范例:

cal : 显示本月的月历。

[root@mylinux /root]# date
Tue Aug 15 08:00:18 CST 2000
[root@mylinux /root]# cal
August 2000
Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

cal 2001 : 显示公元 2001 年年历。
[root@mylinux /root]# cal 2001
2001

January February March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1 2 3
7 8 9 10 11 12 13 4 5 6 7 8 9 10 4 5 6 7 8 9 10
14 15 16 17 18 19 20 11 12 13 14 15 16 17 11 12 13 14 15 16 17
21 22 23 24 25 26 27 18 19 20 21 22 23 24 18 19 20 21 22 23 24
28 29 30 31 25 26 27 28 25 26 27 28 29 30 31

April May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 5 1 2

8 9 10 11 12 13 14 6 7 8 9 10 11 12 3 4 5 6 7 8 9
15 16 17 18 19 20 21 13 14 15 16 17 18 19 10 11 12 13 14 15 16
22 23 24 25 26 27 28 20 21 22 23 24 25 26 17 18 19 20 21 22 23
29 30 27 28 29 30 31 24 25 26 27 28 29 30

July August September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 1
8 9 10 11 12 13 14 5 6 7 8 9 10 11 2 3 4 5 6 7 8
15 16 17 18 19 20 21 12 13 14 15 16 17 18 9 10 11 12 13 14 15
22 23 24 25 26 27 28 19 20 21 22 23 24 25 16 17 18 19 20 21 22
29 30 31 26 27 28 29 30 31 23 24 25 26 27 28 29
30
October November December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1
7 8 9 10 11 12 13 4 5 6 7 8 9 10 2 3 4 5 6 7 8
14 15 16 17 18 19 20 11 12 13 14 15 16 17 9 10 11 12 13 14 15
21 22 23 24 25 26 27 18 19 20 21 22 23 24 16 17 18 19 20 21 22
28 29 30 31 25 26 27 28 29 30 23 24 25 26 27 28 29
30 31

cal 5 2001： 显示公元 2001 年 5 月月历。
[root@mylinux /root]# cal 5 2001
May 2001
Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[root@mylinux /root]#

cal -m： 以星期一为每周的第一天方式，显示本月的月历。
[root@mylinux /root]# cal -m
August 2000
Mo Tu We Th Fr Sa Su
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27

28 29 30 31

cal -jy : 以一月一日起的天数显示今年的年历。
[root@mylinux /root]# cal -jy
2000

January February
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
1 32 33 34 35 36
2 3 4 5 6 7 8 37 38 39 40 41 42 43
9 10 11 12 13 14 15 44 45 46 47 48 49 50
16 17 18 19 20 21 22 51 52 53 54 55 56 57
23 24 25 26 27 28 29 58 59 60
30 31
March April
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
61 62 63 64 92
65 66 67 68 69 70 71 93 94 95 96 97 98 99
72 73 74 75 76 77 78 100 101 102 103 104 105 106
79 80 81 82 83 84 85 107 108 109 110 111 112 113
86 87 88 89 90 91 114 115 116 117 118 119 120
121
May June
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
122 123 124 125 126 127 153 154 155
128 129 130 131 132 133 134 156 157 158 159 160 161 162
135 136 137 138 139 140 141 163 164 165 166 167 168 169
142 143 144 145 146 147 148 170 171 172 173 174 175 176
149 150 151 152 177 178 179 180 181 182

July August
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
183 214 215 216 217 218
184 185 186 187 188 189 190 219 220 221 222 223 224 225
191 192 193 194 195 196 197 226 227 228 229 230 231 232
198 199 200 201 202 203 204 233 234 235 236 237 238 239
205 206 207 208 209 210 211 240 241 242 243 244
212 213
September October
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
245 246 275 276 277 278 279 280 281
247 248 249 250 251 252 253 282 283 284 285 286 287 288

254 255 256 257 258 259 260 289 290 291 292 293 294 295
261 262 263 264 265 266 267 296 297 298 299 300 301 302
268 269 270 271 272 273 274 303 304 305

November December
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
306 307 308 309 336 337
310 311 312 313 314 315 316 338 339 340 341 342 343 344
317 318 319 320 321 322 323 345 346 347 348 349 350 351
324 325 326 327 328 329 330 352 353 354 355 356 357 358
331 332 333 334 335 359 360 361 362 363 364 365
366

360 ☆date

使用权限 : 所有使用者

使用方式 :
date [-u] [-d datestr] [-s datestr] [--utc] [--universal] [--date=datestr] [--set=datestr] [--help] [--version] [+FORMAT] [MMDDhhmm[[CC]YY][.ss]]

说明 :
date 可以用来显示或设定系统的日期与时间，在显示方面，使用者可以设定欲显示的格式，格式设定为一个加号后接数个标记，其中可用的标记列表如下 :

参数:
-d datestr : 显示 datestr 中所设定的时间 (非系统时间)
--help : 显示辅助讯息
-s datestr : 将系统时间设为 datestr 中所设定的时间
-u : 显示目前的格林威治时间
--version : 显示版本编号

时间方面 :
% : 印出 %
%n : 下一行
%t : 跳格
%H : 小时(00..23)
%I : 小时(01..12)
%k : 小时(0..23)
%l : 小时(1..12)
%M : 分钟(00..59)
%p : 显示本地 AM 或 PM

%r : 直接显示时间 (12 小时制，格式为 hh:mm:ss [AP]M)
%s : 从 1970 年 1 月 1 日 00:00:00 UTC 到目前为止的秒数
%S : 秒(00..61)
%T : 直接显示时间 (24 小时制)
%X : 相当于 %H:%M:%S
%Z : 显示时区

日期方面 :
%a : 星期几 (Sun..Sat)
%A : 星期几 (Sunday..Saturday)
%b : 月份 (Jan..Dec)
%B : 月份 (January..December)
%c : 直接显示日期与时间
%d : 日 (01..31)
%D : 直接显示日期 (mm/dd/yy)
%h : 同 %b
%j : 一年中的第几天 (001..366)
%m : 月份 (01..12)
%U : 一年中的第几周 (00..53) (以 Sunday 为一周的第一天的情形)
%w : 一周中的第几天 (0..6)
%W : 一年中的第几周 (00..53) (以 Monday 为一周的第一天的情形)
%x : 直接显示日期 (mm/dd/yy)
%y : 年份的最后两位数字 (00.99)
%Y : 完整年份 (0000..9999)

若是不以加号作为开头，则表示要设定时间，而时间格式为 MMDDhhmm[[CC]YY][.ss]，其中 MM 为月份，DD 为日，hh 为小时，mm 为分钟，CC 为年份前两位数字，YY 为年份后两位数字，ss 为秒数

date '+This date now is =>%x ， time is now =>%X ， thank you !'

date -s 14:36:00
设置时间

date -s 991128
设置日期

显示时间后跳行，再显示目前日期 :
date +%T%n%D

显示月份与日数 :
date +%B %d

显示日期与设定时间(12:34:56) :
date --date 12:34:56

注意 :

当你不希望出现无意义的 0 时(比如说 1999/03/07)，则可以在标记中插入 - 符号，比如说 date +%-H:%-M:%-S 会把时分秒中无意义的 0 给去掉，像是原本的 08:09:04 会变为 8:9:4。另外，只有取得权限者(比如说 root)才能设定系统时间。

当你以 root 身分更改了系统时间之后，请记得以 clock -w 来将系统时间写入 CMOS 中，这样下次重新开机时系统时间才会持续抱持最新的正确值。

361 ☆sleep

名称 : sleep
使用权限 : 所有使用者

使用方式 : sleep [--help] [--version] number[smhd]

说明 : sleep 可以用来将目前动作延迟一段时间

参数说明 :

--help : 显示辅助讯息
--version : 显示版本编号
number : 时间长度，后面可接 s、m、h 或 d
其中 s 为秒，m 为 分钟，h 为小时，d 为日数

例子 :
显示目前时间后延迟 1 分钟，之后再次显示时间 :
date;sleep 1m;date

362 ☆time

名称: time
使用权限: 所有使用者

使用方式: time [options] COMMAND [arguments]

说明: time 指令的用途，在于量测特定指令执行时所需消耗的时间及系统资源等资讯。例如 CPU 时间、内存、输入输出等等。需要特别注意的是，部分资讯在 Linux 上显示不出来。这是因为在

Linux 上部分资源的分配函式与 time 指令所预设的方式并不相同，以致于 time 指令无法取得这些资料。

把计

-o or --output=FILE
设定结果输出档。这个选项会将 time 的输出写入 所指定的文件中。如果文件已经存在，系统将覆写其内容。
-a or --append
配合 -o 使用，会将结果写到文件的末端，而不会覆盖掉原来的内容。
-f FORMAT or --format=FORMAT
以 FORMAT 字串设定显示方式。当这个选项没有被设定的时候，会用系统预设的格式。不过你可以用环境变数 time 来设定这个格式，如此一来就不必每次登入系统都要设定一次。
一般设定上，你可以用
\t
表示跳栏，或者是用
\n
表示换行。每一项资料要用 % 做为前导。如果要在字串中使用百分比符号，就用 。(学过 C 语言的人大概会觉得很熟悉)
time 指令可以显示的资源有四大项，分别是:

Time resources
Memory resources
IO resources
Command info

详细的内容如下:

Time Resources
E 执行指令所花费的时间，格式是:[hour]:minute:second。请注意这个数字并不代表实际的 CPU 时间。
e 执行指令所花费的时间，单位是秒。请注意这个数字并不代表实际的 CPU 时间。
S 指令执行时在核心模式(kernel mode)所花费的时间，单位是秒。
U 指令执行时在使用者模式(user mode)所花费的时间，单位是秒。
P 执行指令时 CPU 的占用比例。其实这个数字就是核心模式加上使用者模式的 CPU 时间除以总时间。

Memory Resources
M 执行时所占用的实体内存的最大值。单位是 KB
t 执行时所占用的实体内存的平均值，单位是 KB

K 执行程序所占用的内存总量(stack+data+text)的平均大小，单位是 KB
D 执行程序的自有资料区(unshared data area)的平均大小，单位是 KB
p 执行程序的自有堆叠(unshared stack)的平均大小，单位是 KB
X 执行程序间共享内容(shared text)的平均值，单位是 KB
Z 系统内存页的大小，单位是 byte。对同一个系统来说这是个常数

IO Resources
F 此程序的主要内存页错误发生次数。所谓的主要内存页错误是指某一内存页已经置换到置换档(swap file)中，而且已经分配给其他程序。此时该页的内容必须从置换档里再读出来。
R 此程序的次要内存页错误发生次数。所谓的次要内存页错误是指某一内存页虽然已经置换到置换档中，但尚未分配给其他程序。此时该页的内容并未被破坏，不必从置换档里读出来
W 此程序被交换到置换档的次数
c 此程序被强迫中断(像是分配到的 CPU 时间耗尽)的次数
w 此程序自愿中断(像是在等待某一个 I/O 执行完毕，像是磁碟读取等等)的次数
I 此程序所输入的文件数
O 此程序所输出的文件数
r 此程序所收到的 Socket Message
s 此程序所送出的 Socket Message
k 此程序所收到的信号 (Signal)数量

Command Info
C 执行时的参数以及指令名称
x 指令的结束代码 (Exit Status)

-p or --portability
这个选项会自动把显示格式设定成为:
real %e
user %U
sys %S
这么做的目的是为了与 POSIX 规格相容。
-v or --verbose
这个选项会把所有程式中用到的资源通通列出来，不但如一般英文语句，还有说明。对不想花时间去熟习格式设定或是刚刚开始接触这个指令的人相当有用。

范例:
利用下面的指令
time -v ps -aux

我们可以获得执行 ps -aux 的结果和所花费的系统资源。如下面所列的资料:
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.4 1096 472 ? S Apr19 0:04 init
root 2 0.0 0.0 0 0 ? SW Apr19 0:00 [kflushd]

root 3 0.0 0.0 0 0 ? SW Apr19 0:00 [kpiod]
  .....
root 24269 0.0 1.0 2692 996 pts/3 R 12:16 0:00 ps -aux

Command being timed: "ps -aux"
User time (seconds): 0.05
System time (seconds): 0.06
Percent of CPU this job got: 68%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.16
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 0
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 238
Minor (reclaiming a frame) page faults: 46
Voluntary context switches: 0
Involuntary context switches: 0
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0

363 ☆reset,tset

名称: reset, tset
使用方法: tset [-IQqrs] [-] [-e ch] [-i ch] [-k ch] [-m mapping] [terminal]

使用说明:

reset 其实和 tset 是一同个命令，它的用途是设定终端机的状态。一般而言，这个命令会自动的从环境变数、命令列或是其它的组态档决定目前终端机的型态。如果指定型态是 ? 的话，这个程式会要求使用者输入终端机的型别。

由于这个程式会将终端机设回原始的状态，除了在 login 时使用外，当系统终端机因为程式不正常执行而进入一些奇怪的状态时，你也可以用它来重设终端机 o 例如不小心把二进位档用 cat 指

令进到终端机，常会有终端机不再回应键盘输入，或是回应一些奇怪字元的问题。此时就可以用 reset 将终端机回复至原始状态。选项说明:

-p
将终端机类别显示在屏幕上，但不做设定的动作。这个命令可以用来取得目前终端机的类别。
-e ch
将 erase 字元设成 ch
-i ch
将中断字元设成 ch
-k ch
将删除一行的字元设成 ch
-I
不要做设定的动作，如果没有使用选项 -Q 的话，erase、中断及删除字元的目前值依然会送到屏幕上。
-Q
不要显示 erase、中断及删除字元的值到屏幕上。
-r
将终端机类别印在屏幕上。
-s
将设定 TERM 用的命令用字串的型式送到终端机中，通常在 .login 或 .profile 中用
范例:
让使用者输入一个终端机型别并将终端机设到该型别的预设状态。
reset ?

将 erase 字元设定 control-h
reset -e ^B

将设定用的字串显示在屏幕上
reset -s
Erase is control-B (^B).
Kill is control-U (^U).
Interrupt is control-C (^C).
TERM=xterm;

364☆mtools

Linux 系统提供了一组称为 mtools 的可移植工具，可以让用户轻松地从标准的 MS-DOS 磁盘上读、写文件和目录。它们对 DOS 和 Linux 环境之间交换文件非常有用。它们是不具备共同的文件系统格式的系统之间交换文件的有力手段。对于一个 MS-DOS 的软盘，只要把软盘放在软驱中，就可以利用 mtools 提供的命令来访问软盘上的文件。

配置文件是/etc/mtools.conf

mtools 的主要命令如下:
mattrib, mbadblocks, mcat, mcd, mcopy, mdel, mdeltree, mdir,mdoctorfat, mdu, mformat, minfo, mlabel, mmd, mmount, mpartition,mrd, mread, mmove, mren, mshowfat, mtoolstest, mtype, mwrite,mzip

mcd 目录名 改变 MSDOS 目录;
mcopy 源文件 目标文件 在 MSDOS 和 Unix 之间复制文件;
mdel 文件名 删除 MSDOS 文件;
mdir 目录名 显示 MSDOS 目录;
mformat 驱动器号 在低级格式化的软盘上创建 MSDOS 文件系统;
rnlabel 驱动器号 产生 MSDOS 卷标;
mmd 目录名 建立 MSDOS 目录;
mrd 目录名 删除 MSDOS 目录;
mren 源文件 目标文件 重新命名已存在的 MSDOS 文件;
mtype 文件名 显示 MSDOS 文件的内容。

这些命令和对应的不加 m 的 MSDOS 命令非常相似。

在 Linux 环境下看 DOS 盘最上层的目录的内容:
$ mdir a:

将 DOS 盘上的文件 panda 复制到当前目录下，并用 ls 命令进行验证。
$ mcopy a:panda.doc

365 ☆打印

在 Linux 下采用假脱机（spooling）打印方法，当用户需要打印一个文件时，该文件并不直接送到打印机，而是送到 spool 目录下，然后由一个负责打印的后台进程把这些数据送入打印机。

Linux 对每台打印机都定义了一个打印缓冲区，打印机守护程序经常扫描打印缓冲区以查看有无要打印的新文件。如果存在，就按先进先出的顺序打印缓冲区中的文件。

Linux 系统除了可以在本地打印机上打印外，还可以通过网络打印机远程打印。

Linux 系统提供了一组有关打印的命令。一般情况下，打印命令使用默认打印机;如果用户定义了 PRINTER 环境变量，打印命令就使用这个变量定义的打印机;另外，用户还可以在命令行上指定要使用的打印机。

对打印而言，有一个非常重要的目录，就是打印缓冲区目录，要打印的数据在被打印之前都集中到这里。通常一台打印机对应一个打印缓冲区目录，这样比较容易管理打印机。

系统使用/var/spool/lpd 作为主打印缓冲区，每个单独的打印机都在主打印缓冲区下有一个与这台打印机同名的目录。因此，名为 ps_nff 的打印机把/var/spool/lpd/ps_nff 作为它的打印缓冲区目录。

366 ☆system-config-printer

配置打印机队列

367 ☆lp

368 ☆lpd

使用权限: 所有使用者

使用方式:lpd [-l] [#port]
lpd 是一个常驻的印表机管理程式，它会根据 /etc/printcap 的内容来管理本地或远端的印表机。/etc/printcap 中定义的每一个印表机必须在 /var/lpd 中有一个相对应的目录，目录中以 cf 开头的文件表示一个等待送到适当装置的印表工作。这个文件通常是由 lpr 所产生。

lpr 和 lpd 组成了一个可以离线工作的系统，当你使用 lpr 时，印表机不需要能立即可用，甚至不用存在。lpd 会自动监视印表机的状况，当印表机上线后，便立即将文件送交处理。这个得所有的应用程式不必等待印表机完成前一工作。

参数:
-l: 将一些除错讯息显示在标准输出上。
#port: 一般而言，lpd 会使用 getservbyname 取得适当的 TCP/IP port，你可以使用这个参数强迫 lpd 使用指定的 port。

范例:
这个程式通常是由 /etc/rc.d 中的程式在系统启始阶段执行。

369 ☆lpq

语法: lpq [-E] [-P 打印机] [-a] [-l] [+interval]

说明
lpq 是缓冲队列检查命令，它通过 lpd 在缓冲区中检查打印文件，报告指定作业的状态或指定用户的所有作业。不带任何参数的 lpq 命令显示现在队列中的任何作业。lpq 命令的显示结果中一个重要的信息就是作业标识号（作业 ID），它标识一个特定的作业。如果用户想取消一个挂起的作业，就必须在命令中指定这个标识号。

参数：
-P 指定一个打印机，否则使用缺省打印机或环境变量 PRINTER 指定的打印机。
-l 打印组成作业的所有文件的信息。
-E 链接到服务器的时候强制加密
-a 显示所有打印机的任务
-l 长格式.对提交的每一个作业，lpq 报告用户名、在队列中的级别、组成作业的文件、作业标识以及总的大小等信息。

lpq -P sp 检查 sp 打印机的状态

370 ☆lpr

使用权限: 所有使用者

使用方式:
lpr [-E] [-P destination] [-# num-copies [-l] [-o option] [-p] [-r] [-C/J/Ttitle] [file(s)]

产生一个假脱机文件,并发送到打印机
lpr 同 lpd 守护进程通讯，lpd 扫描/etc/printcap 文件，查询打印机对应的缓存目录，然后由 lpd 控制打印，将需要打印的数据送到实际打印机上。如果没有指定文件，lpr 就使用标准输入。
lpr 命令是脱机打印命令，该命令将打印作业放到打印缓冲队列中。为 Linux 系统指定的每台打印机都有自己的打印缓冲目录，每个目录中的 minfree 文件指定保存打印文件的磁盘块的数量。

参数:
-m 打印完毕后发送 email。
-E 链接到服务器的时候强制加密
-P Printer 将资料送至指定的印表机 Printer，预设值为 lp。 如果不用此选项，则使用缺省打印机或环境变量 PRINTER 指定的打印机。
-# copies 打印 n 份:1~100
-C name 设置任务名
-J name 设置任务名
-T name 设置任务名
-l
Specifies that the print file is already formatted for the destination and should be sent without filtering. This option is equivalent to "-oraw".
-o option
Sets a job option.
-p
Specifies that the print file should be formatted with a shaded header with the date, time, job name, and page number. This option is equivalent to "-oprettyprint" and is only useful when printing text files.
-r

Specifies that the named print files should be deleted after printing them.

lpr -P lp file1.c file2.c

lpr < install.log
打印 install.log

371 ☆enscript

语法: enscript [-P 打印机名] 文件名
例如:
encsript file3 或 enscript -Psp file3 自 sp 打印机打印文件 file3.

372 ☆cancel

取消打印任务

373 ☆lprm

语法: lprm [-P 打印机名] 用户名或作业编号

-- 将一个工作由印表机贮列中移除 用法

/usr/bin/lprm [P] [file...]

说明
尚未完成的印表机工作会被放在印表机贮列之中，这个命令可用来将常未送到印表机的工作取消。由于每一个印表机都有一个独立的贮列，你可以用 -P 这个命令设定想要作用的印列机。如果没有设定的话，会使用系统预设的印表机。

这个命令会检查使用者是否有足够的权限删除指定的文件，一般而言，只有文件的拥有者或是系统管理员才有这个权限。
删除打印机内的打印作业(用户仅可删除自己的打印作业)

将印表机 hpprinter 中的第 1123 号工作移除
lprm -Phpprinter 1123

将第 1011 号工作由预设印表机中移除
lprm 1011

lprm 命令用于从缓冲队列中删除打印作业，用户可以使用该命令从缓冲队列中删除属于自己的一个或多个打印作业。

lprm 命令的格式为：

lprm [-P printer] [-] [job #] [user…]

命令中各选项的含义如下：

-P 指定一个打印机，否则使用缺省打印机或环境变量 PRINTER 指定的打印机。

- 删除用户所有的打印作业。

user 删除队列中属于用户 user 的作业（只有超级用户可以这样做）。

job # 通过指定作业号#删除某个打印作业，作业号可以通过 lpq 命令得到，如：

$ lpq -l

lst:ken [job #013ucbarpa]

（standard input） 100 bytes

$ lprm 13

例如:
lprm user 或 lprm -Psp user 删除 sp 打印机中用户 user 的打印作业,此时用户名必须为 user.
lprm -Psp 456 删除 sp 打印机上编号为 456 的打印作业.

374 ☆troff

语法: ptroff [-P 打印机名] [-man][-ms] 文件名
例如:
ptroff -Psp -man /usr/man/man1/lpr1 以 troff 格式,自 sp 打印机打印 lpr 命令的使用说明.

375 ☆lpc

lpc -P lp status
显示 lp 打印机的状态

lpc disable
禁止发送作业到打印机

lpc start
从打印队列中重新开始传输

lpc stop
停止打印队列和打印机的通讯

376 ☆lpadmin

lpadmin -p pandalp -u deny:panda,wawa
禁止 panda 和 wawa 对 pandalp 打印机的访问

lpadmin -p pandalp -o job-quota-period=84600 -o job-page-limit=10
每天只能打印 10 页,job-k-limit 可以用 k 字节来限制

377 ☆lpstat

lpstat -v 打印机
查看状态

lpstat -c 类
自动打印机类成员

378 ☆CUPS

cups-lpd 是由 xinetd 控制的

web 控制,端口 631

配置文件是/etc/cups/cups.conf

用浏览器打开 http://localhost:631 设置
一，打开以后，点击上面一栏里的 printers;
二，点击左下角的 add printer;
三，然后它要求你填入你的名字和密码,我填的是 root 用户和 root 的目录;
四，填完后要你加入打印机的名字，地址和描述。我的是 canon pixma ip1000,我填的打印机的名字是 canon,地址是你打印机所处的目录，我的打印机是 usb 接口，所以地址是/dev/usb/lp0,描述你可以不填。
五，接下来，它会要你填写打印机的型号，有下拉菜单，你点你的打印机的型号就是。
六，然后要你选驱动程序，你要是装好了驱动的话，它那上面就会列出来。
七，选好了后，应该就可以打印了。

379 ☆启动 CUPS

chkconfig cups-lpd on

xinetd 中打开

service cups restart

380 ☆cupsd.conf

/etc/cups/cupsd.conf

#ServerName myhost.domain.com
打印服务器名,应该与/etc/cups/client.conf 中的客户计算机上的一致

#ServerAdmin root@your.domain.com
管理员地址

#ServerRoot /etc/cups
文件中没有写绝对路径的,就是在这个目录下

#DataDir /usr/share/cups
分类,字体,帮助文档等

#RequestRoot /var/spool/cups
作业文件存储的路径

#TempDir /var/spool/cups/tmp
所有用户可以写入的
要 chmod +t /tmp

#AccessLog /var/log/cups/access_log
用 CUPS web 管理访问的日志

#ErrorLog /var/log/cups/error_log
错误信息

#PageLog /var/log/cups/page_log
记录打印的每一页

#MaxLogSize 0
到多少就轮转一次

0 为不轮转

LogLevel info
日志等级

#Classification classified
#Classification confidential
#Classification secret
#Classification topsecret
#Classification unclassified
安全输出

#DefaultCharset utf-8
#DefaultLanguage en
字符和语言

#DocumentRoot /usr/share/doc/cups-1.1.22
web 管理工具的目录

#PreserveJobHistory Yes
记录已经结束的作业

#PreserveJobFiles No
记录结束作业的假脱机文件,可以重新打印,直到文件清除

#MaxJobs 500
保留旧打印作业的上限

#MaxCopies 100
最大副本上限

Printcap /etc/printcap
#PrintcapFormat BSD
#PrintcapFormat Solaris
打印机设置

#FilterLimit 0
打印作业的限制

#ServerCertificate /etc/cups/ssl/server.crt
#ServerKey /etc/cups/ssl/server.key
加密支持

#RootCertDuration 300
证书刷新时间

#RemoteRoot remroot
远程访问 cups 的用户名

#User lp
#Group sys
标准的 cups 用户

Port 80
Port 631
Listen hostname
Listen hostname:80
Listen hostname:631
Listen 1.2.3.4
Listen 1.2.3.4:631
监听端口和限制监听的网络,主机

#KeepAlive On
web 始终和 cups 连接

#KeepAliveTimeout 60
连接保持打开的时间

#MaxClients 100
最大用户限制

#MaxClientsPerHost 0
用户多次登录到服务器的时间

#MaxRequestSize 0
作业长度限额

#Timeout 300
超时关闭 cups 连接的时间

#Browsing On
打开网络浏览

#BrowseProtocols cups

浏览服务协议
all - Use all supported protocols.
cups - Use the CUPS browse protocol.
slp - Use the SLPv2 protocol.

#BrowseAddress x.y.z.255
#BrowseAddress x.y.255.255
#BrowseAddress x.255.255.255
#BrowseAddress 255.255.255.255
#BrowseAddress @LOCAL
#BrowseAddress @IF(name)
广播地址

#BrowseInterval 30
刷新打印机列表的时间
0 为新增的打印机不发送给其他

#BrowsePoll address:port
从哪里获得打印机列表

#BrowseTimeout 300
不要低于 BrowseInterval,超时从列表中删除

#BrowsePort 631
默认的 IPP 协议的端口

#BrowseRelay source-address destination-address
#BrowseRelay @IF(src) @IF(dst)
给其他网络提供权限,第二个是其他网络的广播地址

#BrowseAllow address
#BrowseDeny address
默认是接受所有

#BrowseOrder allow,deny
#BrowseOrder deny,allow
决定顺序

例子
<Location /printers/lptest>
Order Deny,Allow
Deny From All

Allow From 127.0.0.1
AuthType None
</Location>

381 ☆client.conf

/etc/cups/client.conf
指向默认的打印服务器

382 ☆CUPS 日志

/var/log/cups 中

383 ☆inetd.conf

/etc/inetd.conf 文件

众所周知，作为服务器来说，服务端口开放越多，系统安全稳定性越难以保证。所以提供特定服务的服务器应该尽可能开放提供服务必不可少的端口，而将与服务器服务无关的服务关闭，比如：一台作为 www 和 ftp 服务器的机器，应该只开放 80 和 25 端口，而将其他无关的服务如：finger auth 等服务关掉，以减少系统漏洞。

inetd，也叫作“超级服务器”，就是监视一些网络请求的守护进程，其根据网络请求来调用相应的服务进程来处理连接请求。inetd.conf 则是 inetd 的配置文件。inetd.conf 文件告诉 inetd 监听哪些网络端口，为每个端口启动哪个服务。在任何的网络环境中使用 Linux 系统，第一件要做的事就是了解一下服务器到底要提供哪些服务。不需要的那些服务应该被禁止掉，最好卸载掉，这样黑客就少了一些攻击系统的机会。查看"/etc/inetd.conf"文件，了解一下 inetd 提供哪些服务。用加上注释的方法（在一行的开头加上#号），禁止任何不需要的服务，再给 inetd 进程发一个 SIGHUP 信号：

- 第一步：把文件的许可权限改成 600。

[root@deep]# chmod 600 /etc/inetd.conf

- 第二步：确信文件的所有者是 root。

[root@deep]# stat /etc/inetd.conf

• 第三步：编辑"inetd.conf"文件（vi /etc/inetd.conf），禁止所有不需要的服务，如：ftp、 telnet、shell、 login、 exec、talk、ntalk、 imap、 pop-2、pop-3、finger、auth，等等。如果你觉得某些服务有用，可以不禁止这些服务。

• 第四步：改变了“inetd.conf”文件之后，别忘了给 inetd 进程发一个 SIGHUP 信号（killall -HUP inetd）。

[root@deep /root]# killall -HUP inetd

• 第五步：为了保证“inetd.conf”文件的安全，可以用 chattr 命令把它设成不可改变。把文件设成不可改变的只要用下面的命令：

[root@deep]# chattr +i /etc/inetd.conf

“i”属性的文件是不能被改动的：不能删除或重命名，不能创建这个文件的链接，不能往这个文件里写数据。只有系统管理员才能设置和清除这个属性，如果要改变 inetd.conf 文件，你必须先清除这个不允许改变的标志：

[root@deep]# chattr -i /etc/inetd.conf

但是对于诸如 sendmail，named，www 等服务，由于它们不象 finger，telnet 等服务，在请求到来时由 inet 守护进程启动相应的进程提供服务，而是在系统启动时，作为守护进程运行的。而对于 redhat linux，提供了一个 linuxconfig 命令，可以通过它在图形界面下交互式地设置是否在启动时运行相关服务。也可以通过命令来设置是否启动时启动某个服务，如：[root@deep]# chkconfig -level 35 named off 。

384 ☆services

/etc/services 文件

端口号和标准服务之间的对应关系在 RFC 1700 “Assigned Numbers”中有详细的定义。

/etc/services”文件使得服务器和客户端的程序能够把服务的名字转成端口号，这张表在每一台主机上都存在，其文件名是“/etc/services”。只有“root”用户才有权限修改这个文件，而且在通常情况下这个文件是没有必要修改的，因为这个文件中已经包含了常用的服务所对应的端口号。为了提高安全性，我们可以给这个文件加上保护以避免没有经过授权的删除和改变。为了保护这个文件可以用下面的命令：

[root@deep]# chattr +i /etc/services

385☆securetty

/etc/securetty 文件

“/etc/securetty”文件允许你规定“root”用户可以从那个 TTY 设备登录。登录程序（通常是“/bin/login”）需要读取“/etc/securetty”文件。它的格式是：列出来的 tty 设备都是允许登录的，注释掉或是在这个文件中不存在的都是不允许 root 登录的。

/etc/inittab 文件

将文件中的一行注释掉可以禁止用 Control-Alt-Delete 关闭计算机。如果服务器不是放在一个安全的地方，这非常重要。

编辑 inittab 文件（vi /etc/inittab）把这一行：

ca::ctrlaltdel:/sbin/shutdown -t3 -r now

改为：

#ca::ctrlaltdel:/sbin/shutdown -t3 -r now

用下面的命令使改变生效：

[root@deep]# /sbin/init q

/etc/rc.d/init.d/

/etc/rc.d/init.d/下的脚本主要包含了启动服务的脚本程序。一般用户没有什么必要知道脚本文件的内容。所以应该改变这些脚本文件的权限。

[root@deep]# chmod -R 700 /etc/rc.d/init.d/*

这样只有 root 可以读、写和执行这个目录下的脚本。

386☆mesg

使用权限 ： 所有使用者

使用方式 : mesg [y|n]

说明 ： 决定是否允许其他人传讯息到自己的终端机介面
对于超级用户，系统的默认值为 n;而对于一般用户系统的默认值为 y。

参数:
y : 允许讯息传到终端机介面上。
n : 不允许讯息传到终端机介面上 。

如果 mesg 后不带任何参数，则显示当前的状态是 y 还是 n

改变目前讯息设定，改成不允许讯息传到终端机介面上 ：
mesg n

387 ☆wall

使用权限 ： 所有使用者

使用方式 ：
wall [message]

使用说明:
这个命令的功能是对全部已登录的用户发送信息
wall 会将讯息传给每一个 mesg 设定为 yes 的上线使用者。当使用终端机介面做为标准传入时,
讯息结束时需加上 EOF (通常用 Ctrl+D)

传讯息"hi" 给每一个使用者 ：
wall hi

用户可以先把要发送的信息写好存入一个文件中，然后输入：
wall < 文件名

388 ☆write

使用权限 ： 所有使用者

使用方式 ：
write user [ttyname]

说明 ： 向系统中某一个用户发送信息。

参数:
user : 预备传讯息的使用者帐号
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息

传讯息给 Rollaend，此时 Rollaend 只有一个连线 :
write Rollaend
接下来就是将讯息打上去，结束请按 ctrl+c

传讯息给 Rollaend，Rollaend 的连线有 pts/2，pts/3 :
write Rollaend pts/2
接下来就是将讯息打上去，结束请按 ctrl+c

注意 : 若对方设定 mesg n，则此时讯席将无法传给对方

389 ☆talk

使用权限 : 所有使用者

使用方式 :
talk person [ttyname]

说明 : 与其他使用者对谈

参数:
person : 预备对谈的使用者帐号，如果该使用者在其他机器上，则可输入 person@machine.name
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息

与现在机器上的使用者 Rollaend 对谈，此时 Rollaend 只有一个连线 :
talk Rollaend

接下来就是等 Rollaend 回应，若 Rollaend 接受，则 Rollaend 输入 `talk jzlee`即可开始对谈，结束请按 ctrl+c

与 linuxfab.cx 上的使用者 Rollaend 对谈，使用 pts/2 来对谈 :
talk Rollaend@linuxfab.cx pts/2

接下来就是等 Rollaend 回应，若 Rollaend 接受，则 Rollaend 输入 `talk jzlee@jzlee.home`即可开始对谈，结束请按 ctrl+c

注意 : 若屏幕的字会出现不正常的字元，试着按 ctrl+l 更新屏幕画面。

390 ☆rlogin

(remote login)

功能说明：远端登入。
语法：rlogin [-8EL][-e <脱离字符>][-l <用户名称>][主机名称或 IP 地址]

补充说明：执行 rlogin 指令开启终端机阶段操作，并登入远端主机。

参数：
-8 允许输入 8 位字符数据。
-e 脱离字符> 设置脱离字符。
-E 滤除脱离字符。
-l 用户名称> 指定要登入远端主机的用户名称。
-L 使用 litout 模式进行远端登入阶段操作。

语法:rlogin 主机名[-l 用户名]

rlogin panda -l user
使用 user 帐号登录到工作站 panda 中

391 ☆rwho

功能说明：查看系统用户。

语法：rwho [-a]

补充说明：rwho 指令的效果类似 who 指令，但它会显示局域网里所有主机的用户。主机必须提供 rwhod 常驻服务的功能，方可使用 rwho 指令。

参数：
-a 列出所有的用户，包括闲置时间超过 1 个小时以上的用户。

392 ☆fwhois

功能说明：查找并显示用户信息。

语法：fwhios [帐号名称]

补充说明：本指令的功能有点类似 finger 指令，它会去查找并显示指定帐号的用户相关信息。不同之处在于 fwhois 指令是到 Network Solutions 的 WHOIS 数据库去查找，该帐号名称必须有在上面注册才能寻获，且名称没有大小写的差别。

393 ☆rsh

(remote shell)

功能说明：远端登入的 Shell。

语法：rsh [-dn][-l <用户名称>][主机名称或 IP 地址][执行指令]

补充说明：rsh 提供用户环境，也就是 Shell，以便指令能够在指定的远端主机上执行。

参数：
-d 使用 Socket 层级的排错功能。
-l<用户名称> 指定要登入远端主机的用户名称。
-n 把输入的指令号向代号为/dev/null 的特殊外围设备。

394 ☆ftp

语法:ftp 主机名 或 ftp ip 地址
必须拥有远程工作站的帐号及密码,才可进行传输工作.

ftp doc 与远程工作站 doc 之间进行文件传输.
Name(doc:user-name):<输入帐号>
Password(doc:user-password):<输入密码>
ftp>help 列出 ftp 文件传输时可使用的命令.
ftp>!ls 列出本地工作站当前目录下的所有文件名
ftp>!pwd 列出本地工作站当前所在的目录位置
ftp>ls 列出远程工作站当前目录下的所有文件名
ftp>dir 列出远程工作站当前目录下的所有文件名
ftp>dir.|more 分页列出远程工作站当前目录下的所有文件名
ftp>pwd 列出远程工作站当前所在的目录位置
ftp>cd dir1 更改远程工作站的工作目录位置至 dir1 之下
ftp>get file1 将远程工作站的文件 file1 拷贝到本地工作站中
ftp>put file2 将本地工作站的文件 file2 拷贝到远程工作站中
ftp>mget *.c 将远程工作站中扩展文件名为 c 的所有文件拷贝到本地工作站中
ftp>mput *.txt 将本地工作站中扩展文件名为 txt 的所有文件拷贝到远程工作站中
ftp>prompt 切换交互式指令(使用 rmput/mget 时不是每个文件皆询问 yes/no)
ftp>quit 结束 ftp 工作
ftp>bye 结束 ftp 工作

FTP 命令是 Internet 用户使用最频繁的命令之一,不论是在 DOS 还是 UNIX 操作系统下 使用 FTP,都会遇到大量的 FTP 内部命令,熟悉并灵活应用 FTP 的内部命令,可以大大方便 使用者,对于现在拨号上网的用户，如果 ISP 提供了 shell 可以使用 nohup，那么 ftp 将是 你最省钱的上 download 方式，ftp 的命令行格式为:ftp -v -d -i -n -g[主机名]

参数:
-v 显示远程服务器的所有响应信息。
-d 使用调试方式。
-n 限制 ftp 的自动登录,即不使用.netrc 文件。
-g 取消全局文件名。

ftp 使用的内部命令如下(其中括号表示可选项):
1.![cmd[args]]在本地机中执行交互 shell、exit 回到 ftp 环境,如!ls＊.zip。
2.￥ macro-ame[args]执行宏定义 macro-name。
3.account[password]提供登录远程系统成功后访问系统资源所需的补充口令。
4.appendlocal-file[remote-file]将本地文件追加到远程系统主机,若未指定远程系统文件名,则使用本地文件名。
5.ascii 使用 ascii 类型传输方式。
6.bell 每个命令执行完毕后计算机响铃一次。
7.bin 使用二进制文件传输方式。
8.bye 退出 ftp 会话过程。
9.case 在使用 mget 时,将远程主机文件名中的大写转为小写字母。
10.cd remote-dir 进入远程主机目录。
11.cdup 进入远程主机目录的父目录。
12.chmod modefile-name 将远程主机文件 file-name 的存取方式设置为 mode,如 chmod 777 a.out。
13.close 中断与远程服务器的 ftp 会话(与 open 对应)。
14.cr 使用 asscii 方式传输文件时,将回车换行转换为回行。
15.delete remote-file 删除远程主机文件。
16.debug[debug-value]设置调试方式,显示发送至远程主机的每条命令,如 debug3,若 设为 0,表示取消 debug。
17.dir[remote-dir][local-file]显示远程主机目录,并将结果存入 local-file。
18.disconnection 同 close。
19.form format 将文件传输方式设置为 format,缺省为 file 方式。
20.getremote-file[local-file]将远程主机的文件 remote-file 传至本地硬盘的 local-file。
21.glob 设置 mdelete、mget、mput 的文件名扩展,缺省时不扩展文件名,同命令行的-g 参数。
22.hash 每传输 1024 字节,显示一个 hash 符号(＃)。
23.help[cmd]显示 ftp 内部命令 cmd 的帮助信息,如 help get。
24.idle[seconds]将远程服务器的休眠计时器设为[seconds]秒。
25.image 设置二进制传输方式(同 binary)
26.lcd[dir]将本地工作目录切换至 dir。
27.ls[remote-dir][local-file]显示远程目录 remote-dir,并存入本地 local-file。

28.macdef macro-name 定义一个宏,遇到 macdef 下的空行时,宏定义结束。
29.mdelete[remote-file]删除远程主机文件。
30.mdir remote-files local-file 与 dir 类似,但可指定多个远程文件,如 mdir＊.o.＊. zipoutfile。
31.mget remote-files 传输多个远程文件。
32.mkdir dir-name 在远程主机中建一目录。
33.mls remote-file local-file 同 nlist,但可指定多个文件名。
34.mode[mode-name]将文件传输方式设置为 mode-name,缺省为 stream 方式。
35.modtime file-name 显示远程主机文件的最后修改时间。
36.mput local-file 将多个文件传输至远程主机。
37.newerfile-name 如果远程机中 file-name 的修改时间比本地硬盘同名文件的时间更近,则重传该文件。
38.nlist[remote-dir][local-file]显示远程主机目录的文件清单,并存入本地硬盘的 local-file。
39.nmap[inpatternoutpattern]设置文件名映射机制,使得文件传输时,文件中的某些字符相互转换,如 nmap￥1.￥2.￥3[￥1,￥2].[￥2,￥3],则传输文件 a1.a2.a3 时 ,文件名变为 a1、a2，该命令特别适用于远程主机为非 U-NIX 机的情况。
40.ntrans[inchars[outchars]]设置文件名字符的翻译机制,如 ntrans1R,则文件名 LL L 将变为 RRR。
41.open host[port]建立指定 ftp 服务器连接,可指定连接端口。
42.passive 进入被动传输方式。
43.prompt 设置多个文件传输时的交互提示。
44.proxyftp-cmd 在次要控制连接中,执行一条 ftp 命令,该命令允许连接两个 ft p
服务器,以在两个服务器间传输文件。第一条 ftp 命令必须为 open,以首先建立两个服务器间的连接。
45.put local-file[remote-file]将本地文件 local-file 传送至远程主机。
46.pwd 显示远程主机的当前工作目录。
47.quit 同 bye,退出 ftp 会话。
48.quote arg1,arg2……将参数逐字发至远程 ftp 服务器,如 quote syst。
49.recv remote-file[local-file]同 get。
50.regetremote-file[local-file]类似于 get,但若 local-file 存在,则从上次传输中断处续传。
51.rhelp[cmd-name]请求获得远程主机的帮助。
52.rstatus[file-name]若未指定文件名,则显示远程主机的状态,否则显示文件状态。
53.rename[from][to]更改远程主机文件名。
54.reset 清除回答队列。
55.restart marker 从指定的标志 marker 处,重新开始 get 或 put,如 restart 130。
56.rmdir dir-name 删除远程主机目录。
57.runique 设置文件名唯一性存储,若文件存在,则在原文件后加后缀。
58.send local-file[remote-file]同 put。
59.sendport 设置 PORT 命令的使用。
60.site arg1,arg2……将参数作为 SITE 命令逐字发送至远程 ftp 主机。
61.size file-name 显示远程主机文件大小,如 site idle 7200。
62.status 显示当前 ftp 状态。
63.struct[struct-name]将文件传输结构设置为 struct-name,缺省时使用 stream 结构。
64.sunique 将远程主机文件名存储设置为唯一(与 runique 对应)。
65.system 显示远程主机的操作系统类型。

66.tenex 将文件传输类型设置为 TENEX 机所需的类型。
67.tick 设置传输时的字节计数器。
68.trace 设置包跟踪。
69.type[type-name]设置文件传输类型为 type-name,缺省为 ascii,如 typebinary,设置二进制传输方式。
70.umask[newmask]将远程服务器的缺省 umask 设置为 newmask,如 umask 3。
71.useruser-name[password][account]向远程主机表明自己的身份,需要口令时,必须输入口令,如 user anonymous my@email。
72.verbose 同命令行的-v 参数,即设置详尽报告方式,ftp 服务器的所有响应都将显示给用户,缺省为 on.
73.?[cmd]同 help。

那么如何应用这些命令提高效率呢？下面我举一个例子，如何利用 fttp 进行后台下载，假设你的 ISP 给你提供了 shell 并且可以用 nohup，你想由 fttp.download.com/pub/internet/ 下载一个 30M 的程序 aaa.zip 具体步骤如下：
1.用 notepad 做一个文件如 aaa1 内容如下
open ftp.dwonload.com
user anonymous zyz@cenpok.net
cd /pub/internet/
i
get aaa.zip
close
bye
2.拨号登录到你的 ISP 上。用 telnet 或 netterm 登录到 shell，一般都在你的 home 子目录里 bbs~/
3.用 fttp 上传 aaa1 到 ISP 服务器你的子目录。
4. 执行 nohup fttp -invd aaa2＆
这样这个进程就被放在 ISP 服务器的后台进行了，如果你想知道情况如何，可以 more aaa2 就可以知道情况如何了。这时你可以断线了或干点别的，估计时间到了(time 约=30M/(33.6K/9)s)拨号上去，more aaa2 如果显示成功下载 aaa.zip 就表示 aaa.zip 已经被下载到 ISP 的服务器上了，你再由 ISP 的服务器拉回来就相当与点对点了,记得下载完成后 del 掉你的文件(aaa.zip)，免得浪费 ISP 资源，它会关掉 shell 的。

395☆lftp

主页：http://lftp.yar.ru/

一、简介：在大多发行版都有打包，请到各大发行版的 ftp 列表中得到，或者在发行版的安装盘中也能得到。lftp 是一个命令行式的 ftp 客户端。对中文支持较好。如果您在 linux 的 text 模式下，要安装 zhcon 或者 cce 之类的。

安装：

1、RPM 包管理的系统，请到

http://freshrpms.net
http;//rpmfind.net 上查找 lftp 的最新包，可以用

#rpm -ivh name.rpm
#rpm -Uvh name.rpm 这是升级之用

2、源码包安装举例：lftp-3.2.0.tar.bz2

#tar zxvf lftp-3.2.0.tar.bz2
#cd lftp-3.2.0
#./configure
#make
#make install

3.调用方法：

lftp ftp://用户名:密码@地址

比如：
[panda@S01~]$lftp ftp://panda@192.169.152.128
口令:
lftp panda@192.169.152.128:~>
lftp panda@192.169.152.128:~>ls

-rw-r--r-- 1 1000 100 44387 May 18 10:04 xvmain.jpg
-rw-r--r-- 1 1000 100 202643 May 18 09:45 xxx.jpeg
-rw-r--r-- 1 1000 100 0 May 20 10:01 鲨鱼的故事.txt

三、使用方法：

0.简单的用法：lcd 切换本地目录，比如 lcd /opt
get 取回一个文件，put 向 ftp 服务器传文件;

1、获得帮助：

代码:

```
lftp panda@192.169.152.128:~> help
        !<shell-command>                        (commands)
        alias [<name> [<value>]]                anon
        bookmark [SUBCMD]                       cache [SUBCMD]
        cat [-b] <files>                        cd <rdir>
        chmod [OPTS] mode file...               close [-a]
        [re]cls [opts] [path/][pattern]         debug [<level>|off] [-o <file>]
        du [options] <dirs>                     exit [<code>|bg]
        get [OPTS] <rfile> [-o <lfile>]         glob [OPTS] <cmd> <args>
        help [<cmd>]                            history -w file|-r file|-c|-l [cnt]
        jobs [-v]                               kill all|<job_no>
        lcd <ldir>                              lftp [OPTS] <site>
        ls [<args>]                             mget [OPTS] <files>
        mirror [OPTS] [remote [local]]          mkdir [-p] <dirs>
        module name [args]                      more <files>
        mput [OPTS] <files>                     mrm <files>
        mv <file1> <file2>                      [re]nlist [<args>]
        open [OPTS] <site>                      pget [OPTS] <rfile> [-o <lfile>]
        put [OPTS] <lfile> [-o <rfile>]         pwd [-p]
        queue [OPTS] [<cmd>]                    quote <cmd>
        repeat [delay] [command]                rm [-r] [-f] <files>
        rmdir [-f] <dirs>                       scache [<session_no>]
        set [OPT] [<var> [<val>]]               site <site_cmd>
        source <file>                           user <user|URL> [<pass>]
        version                                 wait [<jobno>]
        zcat <files>                            zmore <files>
```

如果针对 lftp 的每个命令的帮助，应该是：

lftp panda@192.169.152.128:~> help 命令

比如

代码:

```
lftp panda@192.169.152.128:~> help get
用法: get [OPTS] <rfile> [-o <lfile>]
Retrieve remote file <rfile> and store it to local file <lfile>.
  -o <lfile> specifies local file name (default - basename of rfile)
  -c    continue, reget
```

-E delete remote files after successful transfer
-a use ascii mode (binary is the default)
-O <base> specifies base directory or URL where files should be placed

lftp stationxx
lftp -u student stationxx

只要键入足够有意义的字符，然后按 Tab 键就可以补全
不过这个是也不是万能的
根本的解决方法是，在每一个空格前加转义字符\
比如说
"cd 蜡笔小新全集\ by\ 小彬"

还有 lftp 的 mirror 命令可以很方便的下载一个目录

396 ☆gftp

主页：http://gftp.seul.org/

简介：gftp 基于 gtk 的 ftp 客户端，大家用的也比较多吧，支持中文目录。如果您的 gftp 不能支持中文，请升级版本。

Written in C and has a text interface and a GTK+ 1.2/2.x interface
Supports the FTP, FTPS (control connection only), HTTP, HTTPS, SSH and FSP protocols
FTP and HTTP proxy server support
Supports FXP file transfers (transferring files between 2 remote servers via FTP)
Supports UNIX, EPLF, Novell, MacOS, VMS, MVS and NT (DOS) style directory listings
Bookmarks menu to allow you to quickly connect to remote sites

下载安装：

在各大发行版中，都有 gftp 的打包，可以用各发行版自带的工具来安装。我只说源码包安装：

源码包安装，通用于所有发行版，举例说明：gftp-2.0.18.tar.bz2

代码:
[root@S01ftp]#ls -lh
总用量 1.4M
-rw-r--r-- 1 root root 1.4M 2005-05-21 09:18 gftp-2.0.18.tar.bz2
[root@S01ftp]#tar jxvf gftp-2.0.18.tar.bz2
[root@S01gftp-2.0.18]#./configure

[root@S01gftp-2.0.18]#make
[root@S01gftp-2.0.18]#make install

注：因为 gftp 依赖 gtk，如果不能 make 过去，您应该指定 PKG_CONFIG_PATH

[root@S01gftp-2.0.18]# export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig

调用：

代码:
[root@S01gftp-2.0.18]#gftp

397 ☆rcp

1.拷贝文件或目录至远程工作站
语法: rcp [-r] 源地址 主机名:目的地址
源地址文件名.目录名或路径.
主机名工作站名.目的地址路径名称
例如:
rcp file1 doc:/home/user 将文件 file 拷贝到工作站 doc 路径/home/user 下
rcp -r dir1 doc:/home/user 将目录 dir1 拷贝到工作站 doc 路径/home/user 下

2.自远程工作站,拷贝文件或目录
语法: rcp [-r] 主机名:源地址 目的地址
主机名工作站名
源地址路径名.
目的地址,文件名,目录名或路径
例如:
rcp doc:/home/user/file1 file2 将工作站 doc 路径/home/user 下的文件 file1,拷贝到当前工作站目录下.文件名改为 file2
rcp -r doc:/home/user/dir1 将工作站 doc 路径/home/user 下的目录 dir1,拷贝到当前工作站的目录下,目录名仍为/dir1

398 ☆wget

wget 非交互式的网络文件下载工具

==========

[panda@S01~]$wget --help
GNU Wget 1.9.1，非交互式的网络文件下载工具。
用法： wget [选项]... [url]...

长选项必须用的参数在使用短选项时也是必须的。

启动：
-V, --version 显示 Wget 的版本并且退出。
-h, --help 打印此帮助。
-b, -background 启动后进入后台操作。
-e, -execute=COMMAND 运行‘.wgetrc’形式的命令。

日志记录及输入文件：
-o, --output-file=文件 将日志消息写入到指定文件中。
-a, --append-output=文件 将日志消息追加到指定文件的末端。
-d, --debug 打印调试输出。
-q, --quiet 安静模式(不输出信息)。
-v, --verbose 详细输出模式(默认)。
-nv, --non-verbose 关闭详细输出模式，但不进入安静模式。
-i, --input-file=文件 下载从指定文件中找到的 URL。
-F, --force-html 以 HTML 方式处理输入文件。
-B, --base=URL 使用 -F -i 文件选项时，在相对链接前添加指定的 URL 。

下载：
-t, --tries=次数 配置重试次数（0 表示无限）。
--retry-connrefused 即使拒绝连接也重试。
-O --output-document=文件 将数据写入此文件中。
-nc, --no-clobber 不更改已经存在的文件，也不使用在文件名后
添加 .#（# 为数字）的方法写入新的文件。
-c, --continue 继续接收已下载了一部分的文件。
--progress=方式 选择下载进度的表示方式。
-N, --timestamping 除非远程文件较新，否则不再取回。
-S, --server-response 显示服务器回应消息。
--spider 不下载任何数据。
-T, --timeout=秒数 配置读取数据的超时时间 (秒数)。
-w, --wait=秒数 接收不同文件之间等待的秒数。
--waitretry=秒数 在每次重试之间稍等一段时间 (由 1 秒至指定的 秒数 不等)。
--random-wait 接收不同文件之间稍等一段时间(由 0 秒至 2*WAIT 秒不等)。
-Y, --proxy=on/off 打开或关闭代理服务器。
-Q, --quota=大小 配置接收数据的限额大小。
--bind-address=地址 使用本机的指定地址 (主机名称或 IP) 进行连接。
--limit-rate=速率 限制下载的速率。

--dns-cache=off 禁止查找存于高速缓存中的 DNS。
--restrict-file-names=OS 限制文件名中的字符为指定的 OS (操作系统) 所允许 的字符。

目录：
-nd --no-directories 不创建目录。
-x, --force-directories 强制创建目录。
-nH, --no-host-directories 不创建含有远程主机名称的目录。
-P, --directory-prefix=名称 保存文件前先创建指定名称的目录。
--cut-dirs=数目 忽略远程目录中指定数目的目录层。

HTTP 选项：
--http-user=用户 配置 http 用户名。
--http-passwd=密码 配置 http 用户密码。
-C, --cache=on/off (不)使用服务器中的高速缓存中的数据 (默认是使用的)。
-E, --html-extension 将所有 MIME 类型为 text/html 的文件都加上 .html 扩 展文件名。
--ignore-length 忽略“Content-Length”文件头字段。
--header=字符串 在文件头中添加指定字符串。
--proxy-user=用户 配置代理服务器用户名。
--proxy-passwd=密码 配置代理服务器用户密码。
--referer=URL 在 HTTP 请求中包含“Referer：URL”头。
-s, --save-headers 将 HTTP 头存入文件。
-U, --user-agent=AGENT 标志为 AGENT 而不是 Wget/VERSION。
--no-http-keep-alive 禁用 HTTP keep-alive（持久性连接）。
--cookies=off 禁用 cookie。
--load-cookies=文件 会话开始前由指定文件载入 cookie。
--save-cookies=文件 会话结束后将 cookie 保存至指定文件。
--post-data=字符串 使用 POST 方法，发送指定字符串。
--post-file=文件 使用 POST 方法，发送指定文件中的内容。

HTTPS (SSL) 选项：
--sslcertfile=文件 可选的客户段端证书。
--sslcertkey=密钥文件 对此证书可选的“密钥文件”。
--egd-file=文件 EGD socket 文件名。
--sslcadir=目录 CA 散列表所在的目录。
--sslcafile=文件 包含 CA 的文件。
--sslcerttype=0/1 Client-Cert 类型 0=PEM (默认) / 1=ASN1 (DER)
--sslcheckcert=0/1 根据提供的 CA 检查服务器的证书
--sslprotocol=0-3 选择 SSL 协议;0=自动选择，
1=SSLv2 2=SSLv3 3=TLSv1

FTP 选项：
-nr, --dont-remove-listing 不删除“.listing”文件。

-g, --glob=on/off 设置是否展开有通配符的文件名。
--passive-ftp 使用“被动”传输模式。
--retr-symlinks 在递归模式中，下载链接所指示的文件(连至目录
则例外）。
递归下载：
-r, --recursive 递归下载。
-l, --level=数字 最大递归深度(inf 或 0 表示无限)。
--delete-after 删除下载后的文件。
-k, --convert-links 将绝对链接转换为相对链接。
-K, --backup-converted 转换文件 X 前先将其备份为 X.old。
-m, --mirror 等效于 -r -N -l inf -nr 的选项。
-p, --page-requisites 下载所有显示完整网页所需的文件，例如图像。
--strict-comments 打开对 HTML 备注的严格(SGML)处理选项。

递归下载时有关接受/拒绝的选项：
-A, --accept=列表 接受的文件样式列表，以逗号分隔。
-R, --reject=列表 排除的文件样式列表，以逗号分隔。
-D, --domains=列表 接受的域列表，以逗号分隔。
--exclude-domains=列表 排除的域列表，以逗号分隔。
--follow-ftp 跟随 HTML 文件中的 FTP 链接。
--follow-tags=列表 要跟随的 HTML 标记，以逗号分隔。
-G, --ignore-tags=列表 要忽略的 HTML 标记，以逗号分隔。
-H, --span-hosts 递归时可进入其它主机。
-L, --relative 只跟随相对链接。
-I, --include-directories=列表 要下载的目录列表。
-X, --exclude-directories=列表 要排除的目录列表。
-np, --no-parent 不搜索上层目录。

请将错误报告或建议寄给 <bug-wget@gnu.org>。

wget 的使用形式是：
wget [参数列表] URL
首先来介绍一下 wget 的主要参数：
- -b：让 wget 在后台运行，记录文件写在当前目录下"wget-log"文件中；
- -t [nuber of times]：尝试次数，当 wget 无法与服务器建立连接时，尝试连接多少次
。比如"-t
120"表示尝试 120 次。当这一项为"0"的时候，指定尝试无穷多次直到连接成功为止，这个设置非常有用，当对方服务器突然关机或者网络突然中断的时候，可以在恢复正常后继续下载没有传完的文件；
- -c：断点续传，这也是个非常有用的设置，特别当下载比较大的文件的时候，如果中途意外中断，那么连接恢复的时候会从上次没传完的地方接着传，而不是又从头开始，使

用这一项需要远程服务器也支持断点续传，一般来讲，基于 UNIX/Linux 的 Web/FTP 服务器都支持断点续传；

- -T [number of seconds]：超时时间，指定多长时间远程服务器没有响应就中断连接，开始下一次尝试。比如"-T 120"表示如果 120 秒以后远程服务器没有发过来数据，就重新尝试连接。如果网络速度比较快，这个时间可以设置的短些，相反，可以设置的长一些，一般最多不超过 900，通常也不少于 60，一般设置在 120 左右比较合适；
- -w [number of seconds]：在两次尝试之间等待多少秒，比如"-w 100"表示两次尝试之间等待 100 秒；
- -Y on/off：通过 / 不通过代理服务器进行连接；
- -Q [byetes]：限制下载文件的总大小最多不能超过多少，比如"-Q2k"表示不能超过 2K 字节，"-Q3m"表示最多不能超过 3M 字节，如果数字后面什么都不加，就表示是以字节为单位，比如"-Q200"表示最多不能超过 200 字节；
- -nd：不下载目录结构，把从服务器所有指定目录下载的文件都堆到当前目录里；
- -x：与"-nd"设置刚好相反，创建完整的目录结构，例如"wget -nd http://www.gnu.org"将创建在当前目录下创建"www.gnu.org"子目录，然后按照服务器实际的目录结构一级一级建下去，直到所有的文件都传完为止；
- -nH：不创建以目标主机域名为目录名的目录，将目标主机的目录结构直接下到当前目录下；
- --http-user=username
- --http-passwd=password：如果 Web 服务器需要指定用户名和口令，用这两项来设定；
- --proxy-user=username
- --proxy-passwd=password：如果代理服务器需要输入用户名和口令，使用这两个选项；
- -r：在本机建立服务器端目录结构；
- -l [depth]：下载远程服务器目录结构的深度，例如"-l 5"下载目录深度小于或者等于 5 以内的目录结构或者文件；
- -m：做站点镜像时的选项，如果你想做一个站点的镜像，使用这个选项，它将自动设定其他合适的选项以便于站点镜像；
- -np：只下载目标站点指定目录及其子目录的内容。这也是一个非常有用的选项，我们假设某个人的个人主页里面有一个指向这个站点其他人个人主页的连接，而我们只想下载这个人的个人主页，如果不设置这个选项，甚至--有可能把整个站点给抓下来，这显然是我们通常不希望的；

ü 如何设定 wget 所使用的代理服务器

wget 可以使用用户设置文件".wgetrc"来读取很多设置，我们这里主要利用这个文件来是设置代理服务器。使用者用什么用户登录，那么什么用户主目录下的".wgetrc"文件就起作用。例如，"root"用户如果想使用".wgetrc"来设置代理服务器，"/root/.wgert"就起作用，下面给出一个".wgetrc"文件的内容，读者可以参照这个例子来编写自己的"wgetrc"文件：

```
http-proxy = 111.111.111.111:8080
ftp-proxy = 111.111.111.111:8080
```

这两行的含义是，代理服务器 IP 地址为：111.111.111.111，端口号为：80。第一行指定

HTTP 协议所使用的代理服务器，第二行指定 FTP 协议所使用的代理服务器。

399☆rsync

只传输变化的部分,可以用安全服务传输

rsync -a /mnt/cdrom/* /var/ftp/pub/inst

chkconfig rsync on

rsync -av -e ssh /mnt/cdrom/* panda@ftpserver:var/ftp/pub/inst

rsync --rsh=ssh student@stationxx:get*

如何用 rsync 修复不完整的 Linux 光盘映像文件
1、光盘映像下载过程中的验证码问题存在的原因;
有时是因为下载工具不支持所致使，比如有些下载工具不支持大于 2G 的文件，所以会出现本来是 3G 的文件，结果他就下载了 2G，文件的不完整导致验证码的不一致;另外一方面有时我们选择的下载地址所提供的文件有问题，也会导致下载的 file.iso 与官方所提供的验证码不一致;

2、用 rsync 工具同步解决映像不完整问题;
大家都知道 rsync 最主要的功能就是同步备份和镜像功能，前提是得有 rsync 服务器;我所说的意思是 rsync 并不能同步 ftp 服务器上的内容，我想这个大家应该明白;
目前大多数 Linux 发行版都有 rsync 服务器，我们就可以利用 rsync 服务器上的源来同步不完整的光盘映像;

当然也能用 rsync 下载 rsync 服务器上的文件，我们用 rsync 的同步修复功能来解决我们所面对的问题，说实在的这个功能真的不错;

2.1 寻找 Linux 发行版的 rsync 服务器;
因为 rsync 并不能同步 ftp 和 web 服务器上的映像文件，所以我们根本没有能力用光盘映像所处的 FTP 或 WEB 服务器上的地址;我们得找有类似下面的地址;
rsync://mirrors.kernel.org

只有在 rsync 服务器上找到我们所需要的映像才能行，寻找的办法是去官方发行版所提供的服务器列表上去找;

2.2 rsync 进入服务器的方法;
比如我通过 wget 下载的 FC-5-i386-DVD.iso 有问题，我想用 rsync 来同步解决;所以首先就要找拥有 FC-5-i386-DVD.iso 的 rsync 服务器;于是我到 Fedora.redhat.com 的下载列表中寻找 ，发现 rsync://mirrors.kernel.org 上有这个文件;所以我们先要进入这台服务器;请不要把地址后面的/省略，这样不会列出服务器或服务器目录中的文件;

[root@panda ~]# rsync rsync://mirrors.kernel.org/
mirrors All mirror sites
debian Debian Linux distribution mirror
redhat RedHat mirror
fedora Fedora - RedHat community project
fedora.us fedora.us - Additional stuff for Fedora
centos CentOS - An Enterprise-class Linux distribution
opensuse OpenSUSE - Novell's community project

我们发现在这个服务器上有 fedora 的镜像;所以我们一级一级的进去;执行下面的命令;
[root@panda ~]# rsync rsync://mirrors.kernel.org/fedora/
drwxr-xr-x 4096 2005/02/03 08:05:26 .
drwxr-xr-x 4096 2006/03/17 00:41:43 core
drwxr-sr-x 4096 2006/03/18 03:05:05 extras
... ...

[root@panda ~]# rsync rsync://mirrors.kernel.org/fedora/core/5/i386/iso/
drwxr-xr-x 4096 2006/03/15 13:39:03 .
-rw-r--r-- 3253669888 2006/03/15 12:49:55 FC-5-i386-DVD.iso
-rw-r--r-- 687235072 2006/03/15 12:47:10 FC-5-i386-disc1.iso
-rw-r--r-- 700618752 2006/03/15 12:48:05 FC-5-i386-disc2.iso
-rw-r--r-- 721016832 2006/03/15 12:50:35 FC-5-i386-disc3.iso
-rw-r--r-- 720910336 2006/03/15 12:51:46 FC-5-i386-disc4.iso
-rw-r--r-- 387753984 2006/03/15 12:52:16 FC-5-i386-disc5.iso
-rw-r--r-- 79122432 2006/03/15 12:31:59 FC-5-i386-rescuecd.iso
-rw-r--r-- 671 2006/03/15 13:38:25 SHA1SUM

我们发现了 FC-5-i386-DVD.iso，这时我们要进行下一个过程，同步文件;

2.3 用 rsync 同步文件的语法;
首先我们要进入本地机存放以前下载好，但有问题的 FC-5-i386-DVD.iso 的目录（还是接着前面的例子），然后再执行类似下面的命令;

命令格式如下：
rsync -vzP 映像文件的 rsync 地址 你以前下载下来的文件

比如我以前下载的 FC-5-i386-DVD.iso 的有问题，这时我想用 rsync 同步来修正，所以我把 FC-5-i386-DVD.iso 文件放入了一个名为 fc5iso 的目录中;
[root@panda ~]# cd fc5
[root@localhost fc5]# ls -la
drwxr-xr-x 2 root root 136 2006-03-26 04:05 .
drwxr-xr-x 20 panda panda 472 2006-03-26 03:56 ..

-rwxr-xr-x 1 panda panda 2276458496 2006-03-26 03:47 FC-5-i386-DVD.iso

然后执行;
[root@localhost fc5]#rsync -vzP rsync://mirrors.kernel.org/fedora/core/5/i386/iso/FC-5-i386-DVD.iso FC-5-i386-DVD.iso

可能要等几分钟才相关的信息出现，也会出现下载提示之类的;如果有下载之类的信息，就不必管他了，这说明已经正常工作了;我们有的只是等待;如果我们再看一下这个目录都有什么，我们就明白了。rsync 的原理是先从本地已存的文件中下载，然后对照服务器的文件，如果已经下载的文件不完整，他就从服务器上下载;如果有错误的，就修正;他会先产生一个临时文件，也就类似下面的... ...

[root@localhost fc5]# ls -la
ls -la
总用量 5253889
drwxr-xr-x 2 root root 136 2006-03-26 04:05 .
drwxr-xr-x 20 panda users 472 2006-03-26 03:56 ..
-rwxr-xr-x 1 root root 2276458496 2006-03-26 03:47 FC-5-i386-DVD.iso
-rwx------ 1 root root 3103522816 2006-03-26 08:59 .FC-5-i386-DVD.iso.SI37yU

等完成后，会有类似下面这样的提示，然后 rsync 会自动删除.file 临时文件，更新 FC-5-i386-DVD.iso。

FC-5-i386-DVD.iso
  3253669888 100% 164.78kB/s 5:21:22 (1, 100.0% of 1)
wrote 381831 bytes read 953051797 bytes 48730.35 bytes/sec
total size is 3253669888 speedup is 3.41

经过这个简单的过程，我们就把有问题的文件同步更新了。然后我们再来检测文件的完整性;这样就可以修正存在问题的文件，是不是能行，只有尝试了才知道;

注意：同步文件时，要有大一点的空间;因为同步的过程中，rsync 会创建一个和你下载文件同等大小的临时文件;

400☆md5sum

md5sum filename

401☆gpg

#生成密匙
代码:

gpg --gen-key

＃显示密匙，公匙，私匙
代码:
gpg --list-keys
gpg --list-public-keys
gpg --list-secret-keys

＃以 ascii 形式 dump 出公匙,私匙。
代码:
gpg --export -a userID >public.key
gpg --export-secret-key -a userID>secret.key

＃导入 key.file
代码:
gpg --import key.file

加密生成 signaure 和公匙加密的文件。
代码:
gpg -se -r userID file

#加密生成没有 signaure 的加密文件。
代码:
gpg -e -r userID file

#加密生成 signaure 和 symmetric 形式的加密文件。
代码:
gpg -sc -r userID file

#加密生成没有 signaure 的 symmetric 加密文件。
代码:
gpg -c -r userID file

#加密生成 sig 式的 detachable 文件。
代码:
gpg -b -r userID file

#解密文件
代码:
gpg [--decrypt] file

#复核 sig 文件。

代码:
gpg --verify sig.file

#删除密匙,私匙，私匙和公匙。
代码:
gpg --delete-key userID
gpg --delete-secret-key userID
gpg --delete-secret-and-public-key userID
更多的用法请参看 manpage.

402 ☆gpg 实验

使用 gpg 来交换加密的电子邮件

用户 alice 和 bob 希望能够安全的交换信息，使用 GNU 的 Privacy Guard（gpg）来提供加密服务。您将建立两个用户，为他们的每个建立公钥和私钥。下一步，alice 将获得 bob 的公钥，并且使用该公钥来加密给他的信息，bob 将解密信息。

1. 建立用户

useradd alice; useradd bob
passwd alice
passwd bob

2.为每个用户建立公钥和密钥。注意到 gpg 第一次运行的时候，会建立缺省的初始化用户文件:当提示"…your message"，按下 CTRL-C。
su - alice
mkdir ~/.gnupg
gpg
gpg -help
gpg --gen-key

su - bob
mkdir ~/.gnupg
gpg #"真实的名称"为 bobby
gpg -help
gpg --gen-key

您将被提示多种的密钥参数。选择缺省的选项。当询问您的细节的时候，设定真实的名称为 Alice。其他的信息您随便填写。您同时将被会问及您的密码。您可以使用任何密码（但是您必须自己记得住！）;或者简单的敲两下<回车>不使用密码。

对于用户 bob 采用相同的步骤，设定其"真实的名称"为 bobby（gpg 抱怨 bob 太短了）

3. 检查 alice 的公钥和密钥，该公钥和密钥存储在 pubring.gpg 和 secring.gpg 中。

```
su - alice
ls ~/.gnupg
echo no-secmem-warning >> ~/.gnupg/gpg.conf

gpg --list-keys
/home/alice/.gnupg/pubring.gpg
------------------------------
pub 1024D/168F25D7 2003-09-18 Alice (demo key) <alice@station1.com>
sub 1024g/CE26F831 2003-09-18

gpg --list-secret-keys
/home/alice/.gnupg/secring.gpg
------------------------------
sec 1024D/168F25D7 2003-09-18 Alice (demo key) <alice@station1.com>
ssb 1024g/CE26F831 2003-09-18
```

4.让 bob 把他的公钥放在 ASCII 文件中以便方便的传递给 alice。接下来，让 alice 导入 bob 的公钥到她的公钥环中去。

```
su - bob
gpg --export --armor bobby > /tmp/bob.key
cat /tmp/bob.key

su - alice
gpg --import /tmp/bob.key
gpg --list-keys
/home/alice/.gnupg/pubring.gpg
------------------------------
pub 1024D/168F25D7 2003-09-18 Alice <alice@station1.com>
sub 1024g/CE26F831 2003-09-18

pub 1024D/67C0F0AD 2003-09-18 bobby <bob@ station1.com >
sub 1024g/FDD05A7A 2003-09-18
```

5.现在 alice 获得了 bob 公钥的副本，并且把它加入到她的公钥环中去。她可以给 bob 发送加密的消息。使用下面的命令步骤来使得 alice 发送给 bob 一个加密的/var/log/dmesg 的副本。

```
cp /var/log/dmesg message.txt
gpg --encrypt --armor --recipient bobby message.txt
```

head message.txt.asc

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.0.6 (GNU/Linux)
Comment: For info see http://www.gnupg.org

hQEOAzUJ6BL90Fp6EAP+J1gPH9RHQ1C+CaJGWSzUD2A603nspW2Ab+fQy7rmJbSA
5lwIPe5IzdmgSwMy80aefARQokI/cgdiWpb20Wzy2bltP413j/mrOiworKCOKguH
IJDQPqYxeticJSbwdZoTozsnLmWKp4uxappv3IaSI91w7REgN0KcwVetIn6UsYsE
AIKOqs1oXdYfU3Kzmt3DficQsZDgCuU1mVESCprb7Iyo/TvjjNuc9imqskrSveZZ
vFU8Loc7uI+gQ4HGUpFNryErMbaR2+KQnJCIz9GZJG/Lr7tFND4wCkFsu3jXvN6e
hUl5KRmRV3MWAkdOT4E3ZYF3dOhrdScxnpeIZdL5IDPo0usB9t2ZgIPHp9jKIIAc

mail -s "here it is" bob <message.txt.asc

6.现在让 bob 检查他的邮件，保存 alice 的邮件到文件。如果您不熟悉邮件客户端，您可以使用一个简单的基于命令行的 mail 工具

su - bob

mail

Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/bob": 1 message 1 new
>N 1 alice@stationa Thu Sep 18 22:02 84/4746 "here it is"
& w message_from_alice
"message_from_alice" [New file]
& q

less message_from_alice
gpg message_from_alice

您将被提示作为纯文本文件的文件名。使用缺省的 message.txt。注意 gpg 自动的执行期望的动作，即，使用恰当的私钥进行解密。缺省的行为可以通过命令行的参数进行改变。

403 ☆文本编辑器

按功能它们可以分为两大类：行编辑器（Ed、Ex）和全屏幕编辑器（Vi、Emacs）。
行编辑器每次只能对一行进行操作，使用起来很不方便。
而全屏幕编辑器可以对整个屏幕进行编辑，用户编辑的文件直接显示在屏幕上，修改的结果可以立即看出来，克服了行编辑的那种不直观的操作方式，便于用户学习和使用，具有强大的功能。

404 ☆vi

Vi 是“Visual interface”的简称

405 ☆vi 参数

vi +5 example1.c
第五行开始

vi +/int example1.c
第一个匹配的

406 ☆vi 命令模式

启用 vim 的功能
:edit ~/.vimrc 这是 Unix 系统所使用的命令
:edit $VIM/_vimrc 这是 Windows 系统所使用的命令
:read $VIMRUNTIME/vimrc_example.vim
:write

ex 模式:
Q 进入 ex 模式
vi 回到命令模式
Escape 总是可以回到命令模式
:r filename 在当前光标的位置插入文件
:q 退出
:w 保存
:w filename 另存为
:#,# w FILENAME 保存部分到文件
:x 或:wq 保存退出,直接用 ZZ 一样的
:q! 不保存强制退出
:!外部命令
:help 参数

407 ☆vi 命令语法

[number] d object 或者 d [number] object
number - 代表执行命令的次数(可选项，缺省设置为 1)。
d - 代表删除。
object - 代表命令所要操作的对象
w - 从当前光标当前位置直到单字/单词末尾，包括空格。
e - 从当前光标当前位置直到单字/单词末尾，但是 *不* 包括空格。

$ - 从当前光标当前位置直到当前行末。

408 ☆vi 缓冲控制

:sp filename 在新窗口中打开文件
Ctrl+w,Ctrl+w 在窗口间切换

409 ☆vi 移动

移动字符:
h 左
j 下
k 上
l 右
移动单词:
vi 中按空格或标点符号来分割单词
vi 中按空格或新行来分割长单词
w 下一单词的第一个字符
W 下一长单词的第一个字符
e 下一单词的最后一个字符
E 下一长单词的最后一个字符
b 前一单词的第一个字符
B 前一长单词的第一个字符
移动行:
0 到行首
$ 到行尾
移动句:
(到句首
) 到句尾
移动段:
{ 到段首
} 到段尾
移动页:
Ctrl+F 或 Page Up 到上页
Ctrl+B 或 Page Down 到下页
Ctrl+u 将屏幕向前（文件头方向）翻滚半屏;
Ctrl+d 将屏幕向后（文件尾方向）翻滚半屏。

特定行:
<行号>SHIFT-G 跳到指定行数
SHIFT-G 则直接跳转到文件中的最后一行
^g 显示当前编辑文件中当前光标所在行位置以及文件状态信息

行号+SHIFT-G 跳转
在当前屏幕中
H 跳到第一行
M 跳到中间一行
L 跳到最后一行

410 ☆vi 搜索

正则搜索:
/ 正向搜索
? 反向搜索
/regexp 到文本样式下一次出现的地点
/foo\.gif 注意字符转换匹配 foo.gif
n 向前重复搜索
N 向后重复搜索 shift-n
% 括号匹配,移动光标到括号上

411 ☆vi 文本替换

:s/regexp/replacenemt/ 替换当前行的第一个式样式
:s/regexp/replacenemt/g 替换当前行的所有匹配
:%s/regexp/replacenemt/g 文件中所有匹配样式
:%s/regexp/replacenemt/gc 每次替换要求确认
:行号,行号 s/regexp/replacenemt/g

412 ☆vi 编辑控制

. 重复上一命令
J 将下一行连接到当前行行末
x 删除当前光标所在字符
r 替换原字符
yy 拷贝当前行
dd 删除当前行
dw 从当前位置删除到下一单词词首
d$ 删除到行末
d) 删除到下一句首
d} 删除本段剩余部分
u 恢复一次命令
U 恢复当前行
^R 重做
(数字)(移动命令)

3w 先后 3 个单词
12b 向前 12 个单词
4j 向下 4 行
(编辑)(数字)(移动命令)
d3w 删除光标后 3 个单词
d2j 删除当前行和下俩行
xp 交换两个字符位置
ddp 上下两行调换

插入模式:
Escape 返回命令模式
i 或 a 进入插入编辑模式:
i 在光标处字符之前插入
I 文本插到当前行的行首
a 在光标处字符之后插入
A 文本插到当前行的行末
o 当前行下面生成一空行并插入
O 当前行上面生成一空行
c 进去更改编辑模式
s 替换,删除当前字符后,输入
cc 以一新行替换当前整行
c0 替换当前位置到行首
c$ 替换当前位置到行末
cw 替换单词
cW
ce
c)
c}
shift-R 进入改写模式

413 ☆vi 可视模式

v 进入可视模式
移动后
y 复制
d 剪切
回到命令模式后
P 粘贴到光标之后
p 粘贴到光标之前
自动缩进启用后:
ctrl-d 向左缩进一级
ctrl-t 向右缩进一级

:set autoindent 手动启用自动缩进
:set tabstop=4 很流行
:set ic 设置忽略大小写,在/搜索模式中用到
:set hls is 高亮显示匹配模式
:set nohl 取消高亮
:set number 行号
:syn on

414 ☆Linux Shell

415 ☆Terminals

终端（Terminals）
Unix 是可以在许多种机器上运行的操作系统，但人们又如何使用这些机器呢？他们是通过哑终端来连接到这些机器，也就是用键盘、显示器及足够的 electronics （电子元件）组成的机器与中央计算机(central computer)相连。在这些终端上，用户可以敲字符（teletypy），这就是字符串'tty'表示终端设备文件，和'getty'命令的名称来历。

为了避免这些混乱，就创建了一个含有所有不同终端特性的(capability)文件，这就是'termcap'。用一个工具打开'/etc/termcap'瞧瞧，可别吓着了 ;-) 。

Linux 终端大多数用'vt100'或'linux'作为终端类型。

416 ☆xterms

在八十年代初期，产生了一个 Unix 的图形子系统-- the X Window System 。九十年代早期，为了更好地实现基于 Intel 的 Unix 类系统上（如 FreeBSD、NetBSD、Linux）的应用，产生了一个系统分支-- XFree86 。

X Window 中一个很大的好处是可以运行多个虚拟(virtual)终端。甚至在 X Window 下就有这么个应用程序--'xterm'。您将发现'xterm'和'virtual terminal'在很多情况下都是一样的。有的地方说'打开一个 xterm'，其实您不是非要用'xterm'程序，其他的终端模拟器(terminal emulator)，如 rxvt、konsole、aterm、eterm、wterm 等等，一样有效。

终端模拟器（又称为虚拟终端）通过伪(pseudo) tty 设备-- pty 与系统相连，并且使用自己的显示标准-- xterm 。这导致不同的终端模拟器可能在一些按键或程序上存在细小的差别，这取决于模拟器多大程度上遵守了'xterm'的显示标准。

417 ☆Shells

为了在终端中运行程序，需要 shell 。shell 是操作系统的一部分，用来与用户打交道，并且可以用来协调各个命令。

第一个真正的 Unix shell -- 'sh'，亦称为'Bourne shell'，诞生于 1975 年，作者是 Steve Bourne 。很快，出现了其他 shell ，如基于原始'Bourne shell'的'ksh'、'zsh'，后者常用作专属 Unixes 系统中的标准 shell ;也有一些从 C 语言中衍生出来的 shell ，如'csh'或'tcsh'。

在 Linux 中，标注的 shell 是'bash'，即 the GNU Bourne-Again Shell （有点玩笑的味道……）。这个 shell 功能非常强大（甚至有人觉得太庞大了），压缩的 man page 就有 50 KB 。

418 ☆shell 特性

当您敲 ENTER 时，光标(cursor)在哪里并不要紧，因为 shell 总是会整行地读取。

419 ☆自动补齐

<TAB>
这称为'命令行自动补齐'(automatic command line completion)

当输入命令，目录或者是文件名的时候按 [Tab] 键。系统就会帮你补齐可能要输入的东西，如果有多个选择系统会列表出来。你可以看清楚之后再多输入一个或多个 charactor ，再按[Tab]。

420 ☆命令行的历史记录

通过按向上方向键，您可以向后遍历近来在该控制台下输入的命令。用向下方向键可以向前遍历。与 SHIFT 键连用的话，您还可以遍历以往在该控制台中的输出。您也可以编辑旧的命令，然后再运行。

[Ctrl + r]
出现提示：
(reverse-i-search)`':
可以输入命令中的任意一部分,此时你尝试一下输入你以前输入过的命令，当你每输入一个字符的时候，终端都会滚动显示你的历史命令。当显示到你想找的合适的历史命令的时候，直接[Enter]，就执行了历史命令。

[Ctrl + p] 或 [Ctrl + n]
快速向前或向后滚动查找一个历史命令，对于快速提取刚刚执行过不久的命令很有用。

[Ctrl + c]
取消本次命令输入：

这个快捷键可以使你从一个可能你已经厌烦了的命令中安全地退出！！也许是个不值一提的小技巧，但是经验告诉我它很有用。很多 Unix 初学者会习惯性地按 [Enter] 以摆脱困境，但是说不定就会发生灾难性的事件，譬如删除了一个重要的配置文件

421 ☆编辑命令行

通过光标和功能键（Home、End 等键），您可以浏览并编辑命令行，如果您需要，还可以用键盘的快捷方式

光标跳转快捷键：
[Ctrl + a] 跳转至命令行首 Ahead of line
[Ctrl + e] 跳转至命令行尾 End of line
[Ctrl + f] 向前跳转一个字符 jump Forward one character
[Ctrl + b] 向后跳转一个字符 jump Backward one character
[Alt + f] 向前跳转到下一个字的第一个字符
[Alt + b] 向后跳转到下一个字的第一个字符

编辑命令的快捷键：
[Ctrl + w] 向后删除一个字，用来对付刚刚输入的错误字很有用
[Ctrl + u] 从光标当前位置删除所有字符至行首
[Ctrl + k] 从光标当前位置删除所有字符至行尾
[Ctrl + d] 删除光标当前位置的字符
[Ctrl + y] 粘贴最后一个被删除的字
[Alt + d] 删除从光标当前位置，到当前字的结尾字符

<!$>：重复前一个命令最后的参数。

422 ☆命令的排列

command1 ; command2
先执行 command1 ，不管 command1 是否出错，接下来执行 command2 。

command1 && command2
只有当 command1 正确运行完毕后，才执行 command2 。

423 ☆命令的任务调度

当您在终端里运行一个命令或开启一个程序时，终端要等到命令或程序运行完毕后，才能再被使用。在 Unix 中，我们称这样的命令或程序在前台(foreground)运行。如果您想在终端下运行另一个命令，则需要再打开一个新的终端。

但这里还有一个更优雅的办法，称为任务调度(jobbing)或后台(backgrounding)。当您运用任务的调度或将命令置于后台，终端就立即解放了，这样一来，终端立即就可以接受新的输入。为实现这样的目的，您只需在命令后面添加一个 & ：
command &

命令 jobs 将告诉您，在这个终端窗口中，运行着哪些命令与程序：
jobs
[1]+ Running command &

当您要关闭终端窗口时，这一点就很重要，因为关闭终端将导致所有在其中运行的任务都将被中止

将前台运行的一个程序放到后台去
command
<CTRL z>
[2]+ Stopped command
组合键<CTRL z>将挂起终端中正在运行的程序

bg
[2]+ command &
用 bg 命令将其放到后台去执行。

424 ☆命令的替换

命令替换(Command substitution)是一项很实用的功能。

这种机制的语法是：
command1 $(command2)

除了'$( )'，您还可以用后引号(backquote)：
command1 `command2`

也可以在几个命令行中输入一个命令，用反斜杠将一个命令行持续到下一行。
$ cp -i \
mydata \
newdata

425 ☆文件名匹配

文件名匹配使得您不必一一写出名称，就可以指定多个文件。您将用到一些特殊的字符，称为通配符(wildcards)。

*
可匹配一个或多个字符。

?
在匹配时，一个问号只能代表一个字符

和?只在方括号外面是通配符，若出现在方括号之内，它们也失去通配符的能力，成为普通字符了。例如，模式"- a[?]abc"中只有一对方括号是通配符，*和?均为普通字符，因此，它匹配的字符串只能是- a*abc 和- a?abc。

[!字符]
您必须将取反号(negation sign)与取反字符放到括号中，不然的话，shell 会将惊叹号(exclamation mark)解释成历史记录替换的开始(the beginning of a history substitution)。

请注意：通配符'*'与取反号连用，很容易产生问题。

rm *[!6]*.bak

表示什么？这个命令将删除所有文件，甚至包括名称中包含'6'的文件。如果您将通配符'*'放到了取反号前面和后面，实际上取反号将失效，因为 shell 将其解释为"所有名称中任何位置都不含该字符的文件"。在我们的例子里，只有文件'666.bak'不符合该模式。

[字符字符]
是个字符集合,必须用中括号将匹配的模式括起来

[字符-字符]
是个字符集合的范围

/
您可以通过反斜线(back slash)来引用特殊字符，比如 ！、$ 、？ 或空格：

或者用（单）引号：
ls '!'*
输出
!56.bak

在 shell 中引号分为三种：单引号，双引号和反引号。

单引号 '
由单引号括起来的字符都作为普通字符出现。特殊字符用单引号括起来以后，也会失去原有意义，而只作为普通字符解释。例如：
$ string='$PATH'

$ echo $string
$PATH

可见$保持了其本身的含义，作为普通字符出现。

双引号 "
由双引号括起来的字符，除$、\、'、和"这几个字符仍是特殊字符并保留其特殊功能外，其余字符仍作为普通字符对待。对于$来说，就是用其后指定的变量的值来代替这个变量和$;对于\而言，是转义字符，它告诉 shell 不要对其后面的那个字符进行特殊处理，只当作普通字符即可。可以想见，在双引号中需要在前面加上\的只有四个字符$，\，'和"本身。而对"号，若其前面没有加\，则 Shell 会将它同前一个"号匹配。
例如，我们假定 PATH 的值为.:/usr/bin:/bin，输入如下命令：
$ TestString＝"$PATH\\\"\$PATH"
$ echo $TestString
  .:/usr/bin:/ bin\"$PATH
$
读者可以自己试一下在第二个双引号之前不加\会产生什么结果。

反引号 `
反引号括起来的字符串被 shell 解释为命令行，在执行时，shell 首先执行该命令行，并以它的标准输出结果取代整个反引号（包括两个反引号）部分。例如：
$ pwd
/home/xyz
$ string="current directory is `pwd`"
$ echo $string
current directour is /home/xyz
$
shell 执行 echo 命令时，首先执行`pwd`中的命令 pwd，并将输出结果/home/xyz 取代`pwd`这部分，最后输出替换后的整个结果。

利用反引号的这种功能可以进行命令置换，即把反引号括起来的执行结果赋值给指定变量。例如：
$ today=`date`
$ echo Today is $today
Today is Mon Apr 15 16:20:13 CST 1999
$

反引号还可以嵌套使用。但需注意，嵌套使用时内层的反引号必须用反斜线（\）将其转义。例如：
$ abc=`echo The number of users is \`who| wc-l\``
$ echo $abc
The number of users is 5
$

在反引号之间的命令行中也可以使用 shell 的特殊字符。Shell 为得到``中命令的结果，它实际上要去执行``中指定的命令。执行时，命令中的特殊字符，如$，"，?等又将具有特殊含义，并且``所包含的可以是任何一个合法的 Shell 命令，如：

```
$ ls
note readme.txt Notice Unix.dir
$ TestString＝"`echo $HOME ` ` ls [nN]*`"
$ echo $TestString
/home/yxz note Notice
$
```

其他情况，读者可自行试之。

注释符

在 shell 编程中经常要对某些正文行进行注释，以增加程序的可读性。在 Shell 中以字符"#"开头的正文行表示注释行。

426 ☆管道

在多个命令间重定向,这要通过管道(pipe)，由管道符号丨来标识。语法是：

command1 | command2 | command3 等等

427 ☆重定向

有时，您希望将命令的输出结果保存到文件中，或以文件内容作为命令的参数。这可以通过'>'和'<'来实现。

command > file

将 command 的输出保存到 file 中，这将覆盖 file 中的内容：

command < file

将 file 内容作为 command 的输入：

一种特殊的方式是'command 2> file'。这将 command 执行的出错信息送到 file 中

另一种操作符是'>>'，这将输出添加到已存在的文件中：

echo "string" >> file

将 string 加到文件 file 中。这是不打开文件而完成编辑的好办法!

但是，'<'和'>'操作符都有一个重要的限制：

command < file1 > file1

将删除 file1 的内容

command < file1 >> file1
将加工过的 file1 内容加回到文件中。

428 ☆bash 配置文件

在您的 home 目录下，运行
[root@panda ~]# ls -al .bash*
-rw------- 1 root root 18086 8 月 23 22:44 .bash_history
-rw-r--r-- 1 root root 24 2004-09-23 .bash_logout
-rw-r--r-- 1 root root 191 2004-09-23 .bash_profile
-rw-r--r-- 1 root root 176 2004-09-23 .bashrc

请注意后两个的区别：'.bash_profile'只在会话开始时被读取一次，而'.bashrc'则每次打开新的终端（如新的 xterm 窗口）时，都要被读取。按照传统，您得将定义的变量，如 PATH ，放到'.bash_profile'中，而象 aliases（别名）和函数之类，则放在'.bashrc'。但由于'.bash_profile'经常被设置成先读取'.bashrc'的内容，您如果图省事的话，就把所有配置都放进'.bashrc'。

这些文件是每一位用户的设置。系统级的设置存储在'/etc/profile'、'/etc/bashrc'及目录'/etc/profile.d'下的文件中。但您得习惯用各自的配置文件：编辑不需要'root'权限，还可以使您的设置更有个性。当系统级与用户级的设置发生冲突时，将采用用户的设置。

读取'.bashrc'的内容，您如果要省点事的话，就把您所有的配置都放进'.bashrc'。

429 ☆bashrc

/etc/bashrc
系统级别上
用于定义别名,函数,umask 和命令行的提示符

430 ☆profile

/etc/profile
系统级别上
用于定义环境,启动文件,相关变量,最后会运行/etc/profile.d 中的脚本

431 ☆profile.d

/etc/profile.d/
缺省的 shell 的定义脚本,在这里

432☆.bash_history

记录了您以前输入的命令

433☆.bash_logout

当您退出 shell 时，要执行的命令

434☆.bash_profile

当您登入 shell 时，要执行的命令

435☆.bashrc

每次打开新的 shell 时，要执行的命令。

436☆提示符

每次当您打开一个控制台(console)或 xterm 时，最先看到的就是提示符(prompt)，类似于：
account@hostname ~ $

在默认设置下，提示符将显示您的用户名、主机名（默认是'localhost'）、当前所在目录（在 Unix 中，'~'表示您的 home 目录）。

按照传统，最后一个字符可以标识您是普通用户（$），还是'root'（#）。

您可以通过 $PS1 变量来设置提示符。命令
echo $PS1

下面是 bash 可识别的全部专用序列的完整列表(您可以在 bash man page 的 "PROMPTING" 部分找到这个列表):
序列 说明
\a ASCII 响铃字符(也可以键入 \007)
\d "Wed Sep 06" 格式的日期
\e ASCII 转义字符(也可以键入 \033)
\h 主机名的第一部分(如 "mybox")
\H 主机的全称(如 "mybox.mydomain.com")
\j 在此 shell 中通过按 ^Z 挂起的进程数
\l 此 shell 的终端设备名(如 "ttyp4")
\n 换行符

\r 回车符
\s shell 的名称(如 "bash")
\t 24 小时制时间(如 "23:01:01")
\T 12 小时制时间(如 "11:01:01")
\@ 带有 am/pm 的 12 小时制时间
\u 用户名
\v bash 的版本(如 2.04)
\V Bash 版本(包括补丁级别)?/td>
\w 当前工作目录(如 "/home/drobbins")
\W 当前工作目录的"基名 (basename)"(如 "drobbins")
\! 当前命令在历史缓冲区中的位置
\# 命令编号(只要您键入内容，它就会在每次提示时累加)
\$ 如果您不是超级用户 (root)，则插入一个 "$";如果您是超级用户，则显示一个 "#"
\xxx 插入一个用三位数 xxx(用零代替未使用的数字，如 "\007")表示的 ASCII 字符
\\ 反斜杠
\[这个序列应该出现在不移动光标的字符序列(如颜色转义序列)之前。它使 bash 能够正确计算自动换行。
\] 这个序列应该出现在非打印字符序列之后。

原文
\a an ASCII bell character (07)
\d the date in "Weekday Month Date" format (e.g., "Tue May 26")
\D{format} the format is passed to strftime(3) and the result is inserted into the prompt string; an empty format results in a locale-specific time representation. The braces are required
\e an ASCII escape character (033)
\h the hostname up to the first '.'
\H the hostname
\j the number of jobs currently managed by the shell
\l the basename of the shell's terminal device name
\n newline
\r carriage return
\s the name of the shell, the basename of $0 (the portion following the final slash)
\t the current time in 24-hour HH:MM:SS format
\T the current time in 12-hour HH:MM:SS format
\@ the current time in 12-hour am/pm format
\A the current time in 24-hour HH:MM format
\u the username of the current user
\v the version of bash (e.g., 2.00)
\V the release of bash, version + patch level (e.g., 2.00.0)
\w the current working directory, with $HOME abbreviated with a tilde
\W the basename of the current working directory, with $HOME abbreviated with a tilde
\! the history number of this command

\# the command number of this command
\$ if the effective UID is 0, a #, otherwise a $
\nnn the character corresponding to the octal number nnn
\\ a backslash
\[begin a sequence of non-printing characters, which could be used to embed a terminal control sequence into the prompt
\] end a sequence of non-printing characters

在 bash 下，可以通过更改 PS1 环境变量的值来设置提示行
$ export PS1="> "
>
更改会立即生效.

默认是
$ export PS1="[\u@\h \W]\$"

通过将 "export" 定义放在您的 ~/.bashrc 文件中可将这种更改固定下来。

标准 Linux 终端和 X 终端允许您设置前景(文字)颜色和背景颜色，如果需要，还可以启用 "bold" 字符。有八种颜色可供我们选择。

颜色是通过在 PS1 中添加专用序列来选择的 -- 基本上是夹在 "\e["(转义开方括号)和 "m" 之间数字值。如果指定一个以上的数字代码，则用分号将它们分开。

下面是一个颜色代码示例:
"\e[0m"
如果将数字代码指定为零，则它就会通知终端将前景、背景和加粗设置重置为它们的默认值。您可能会在在提示行结束时使用这个代码，以使您键入的文字成为非彩色的。

颜色表
首先请查找您要使用的颜色，然后查找对应的前景编号 (30-37) 和背景编号 (40-47)。

例如，如果您喜欢黑底绿字，则可将编号分别设为 32 和 40。 然后打开您的提示行定义并在其中添加适当的颜色代码。下面的定义:
export PS1="\w> "
变为:
export PS1="\e[32;40m\w> "

在 bash 显示出工作目录以后，我们需要使用 "\e[0m" 序列将颜色重新设置为正常值。
export PS1="\e[32;40m\w> \e[0m"

我们还需要将全部非打印字符用专用的 bash 转义序列 "\[" 和 "\]" 括起来。这两个序列通知 bash，被括起来的字符不占用行上的任何空间， 这样就使自动换行能够继续正常工作。没有这两个转义序列，尽管您有了一个非常漂亮的提示行，但是如果您键入的命令恰好到达终端的最右端，就会造成显示混乱。

下面是我们最终的提示行:
export PS1="\[\e[32;1m\]\w> \[\e[0m\]"
别担心在同一个提示行中使用几种颜色，就像下面这样:
export PS1="\[\e[36;1m\]\u@\[\e[32;1m\]\H> \[\e[0m\]"

可以通过在提示行
中添加专用代码来使 X 终端(如 rxvt 或 aterm)的标题栏得到动态更新。 您所要做的只是将下面的序列添加到您的 PS1 提示行中:
"\e]2;titlebar\a"
只须用您希望其出现在 xterm 标题栏中的文字替换子串 "titlebar" 即可

将用户名、主机名和当前工作目录显示在标题栏中，并定义了一个简短、明亮的绿色提示行:
export PS1="\[\e]2;\u@\H \w\a\e[32;1m\]>\[\e[0m\] "
它将全部信息显示在标题栏上，而不是显示在终端上，终端对一行可以显示多少字符有限制。

确保用 "\[" 和 "\]" 将您的标题栏序列括起来(因为就终端而言，这个序列是非打印序列)。
将大量信息放在标题栏中的问题是，如果您使用非图形终端(如系统控制台)，则看不到这些信息。为了解决这个问题，可以在您的 .bashrc 中添加以下几行:

```
if [ "$TERM" = "linux" ]
then
#we're on the system console or maybe telnetting in
export PS1="\[\e[32;1m\]\u@\H > \[\e[0m\]"
else
#we're not on the console, assume an xterm
export PS1="\[\e]2;\u@\H \w\a\e[32;1m\]>\[\e[0m\] "
fi
```

这个 bash 条件语句将根据当前的终端设置动态设置提示行。为了获得一致性，您一定希望配置您的 ~/.bash_profile，以便它在启动时搜索 (source) 您的 ~/.bashrc。确保您的~/.bash_profile 文件中有以下这样一行:
source ~/.bashrc

437 ☆$PATH

'$PATH'与'$PS1'一样，也是环境变量。输入
set

将列出所有当前定义的环境变量。

您看到的这些环境变量在 shell 的配置文件中定义，可能是用户自己的配置文件，也可能是由'root'通过'/etc'下面的系统级文件定义的。如果您使用 X ，更多的一些变量将由 X 、您的窗口管理器或桌面环境的启动文件配置。

首先，作为惯例，所有环境变量名都是大写。由于 Linux 区分大小写，这点您要留意。当然，您可以自己定义一些变量，如'$path'、'$pAtH'，但 shell 不会理睬这些变量。

第二点是变量名有时候以'$'开头，但有时又不是。

当设置一个变量时，您直接用名称，而不需要加'$'：
PATH=/usr/bin:/bin:/usr/local/bin:/usr/X11R6/bin

要获取变量值的话，就要在变量名前加'$'：
echo $PATH
/usr/bin:/bin:/usr/local/bin:/usr/X11R6/bin

否则的话，变量名就会被当作普通文本了：
echo PATH
PATH

要定义一个全局变量，使在以后打开的终端中生效，您需要将局部变量输出(export)，可以用'export'命令：
export PATH=$PATH:/some/directory

现在如果您打开一个新的终端，输入 echo $PATH ，也能看到新设置的 $PATH 了。请注意，命令'export'只能改变当前终端及以后运行的终端里的变量。对于已经运行的终端没有作用。

为了将目录永久添加到您的 $PATH ，只要将'export'的那行添加到您的'.bash_profile'文件中。

请不要在'.bashrc'中设置 PATH ，否则会导致 PATH 中目录的意外增长。您每次打开一个新的 shell ，'.bashrc'都会作用。所以如果在该文件中添加目录，您每次打开一个终端，目录又会被添加。这将导致 PATH 变量由于目录复制，不断地增长。

438 ☆命令的别名

定义别名的语法是：
alias shortcut='command'

命令中有空格的话，就需要用引号（如在命令与可选项间就有空格）。请注意，您可以用单引号或双引号，但他们是有区别的。

单引号将剥夺其中的所有字符的特殊含义，而双引号中的'$'（参数替换）和'`'（命令替换）是例外。这意味着，如果您想在别名中应用变量或命令的替换，就得用双引号。

您可以用'alias'在命令行快速地创建别名，或将命令放到各自的'~/.bashrc'，或放到系统级的'/etc/profile.d/alias.sh'中。要删除一个别名，只要输入：unalias alias 。运行 alias 将列出您系统中所有定义的别名。

如果看一下'~/.bashrc'和'/etc/profile.d/alias.sh'，您会发现系统已经定义了一些别名。

439 ☆Shell 函数

事实上，shell 函数属于 shell 脚本，但可以在同一 shell 下被预加载(preload)和执行（而一般的 shell 脚本至少要打开一个 sub-shell）。

通过 shell 函数，您可以做很多 aliases 无法完成的事情。下面就是一个例子：
function apros() { apropos $1 | egrep -v '(3|\(n\)'; }

定义了一个新命令，称为'apros'。apros name 将先执行'apropos name'（即在 man page 中搜索命令），然后将得到的输出送到管道（|），接着用'egrep'过滤，排除第'3'和第'n'章节的 man page ，这个命令可能没什么大用处，但可以整理'apropos'命令的输出。

函数允许您在函数内部任何位置，使用运行时的参数。而别名，则只允许在命令行尾放一个参数（比如前面的别名'rpmq'）。

'$1'就是位置参数(positional parameter)，表示函数第一个参数的位置标识符。依此类推，还有'$2'等。
function apros() { apropos $1 | egrep -v "\($2"; }

如果您这样运行'apros'命令：
apros name man_section_number

这个命令将搜索标题中含 name 的 man pages ，但排除 man_section_number 部分：
apros menu 3

将搜索标题含'menu'的 man page ，但排除第三章节（关于编程的）。注意到您得引用（quote） 两次，而且还用到了双引号：

您必须引用'egrep'的搜索模式，这样可以不至于被 shell 误解。
您必须用双引号，这样第二个参数才能被正确解释。
您必须引用圆括号，这样使'egrep'按字面意思对待对待参数。

shell 函数的处理类似于别名：将其放到您的'.bashrc'文件，这样就能永久生效了。

440 ☆locale

显示系统的语言环境情况.显示 LANG 和 LC_俩组变量的值
会影响命令输出(从 date 看)

locale -a
显示所有语言支持

从上面得到,设置变量
LANG=zh_CN.GB2312

en_US.utf8

date 的格式不同

/etc/sysconfig/i18n
上面的变量都在这个文件里,文件存储语言变量

LANGUAGE, LC_ALL, LC_MESSAGES, LANG
关于这几个环境变量
一般认为它们的作用是用来指定程序用户界面语言
而且这几个环境变量的优先级是从左到右依次降低的

441 ☆export

用 export PATH 来设置环境变量
比如把下面一行加入到用户家目录下的.bashrc 或.profile 文件中;
export PATH=".:/bin:/sbin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/X11R6/bin"

442 ☆source

source .bashrc

.bashrc 中加一行：
export PATH=".:/bin:/sbin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/X11R6/bin"
有些命令即便能调用，也不会出现任何信息，或者要输入 root 密码的，说明要以超级权限运行的;

443 ☆history

设定命令记录表的长度

语法:set history=n
例如:
set history=40 设定命令记录表的长度为 40(可记录执行过的前面 40 个命令)

查看命令记录表的内容
语法:history

语法:!n
n:命令记录表的命令编号
语法:!string 重复执行前面执行过的以 string 为起始字符串的命令
例如:!cat 重复执行前面执行过的以 cat 为起始字符串的命令

语法:^oldstring^newstring 将前一个命令中 oldstring 的部分改成 newstring 并执行
例如:
find .-name file.c -print
^file.c^core
find.-name core -print

444 ☆alias

语法:alias 查看自己目前定义的所有命令,及所对应的别名.
$ alias

语法:alias name 查看指定的 name 命令的别名.
例如:
alias dir 查看别名 dir 所定义的命令

定义命令的别名
语法:alias name 'command line'
例如:
alias dir 'ls -l'定义别名为 dir

$ alias cls='clear'
在当前会话有效

445 ☆unalias

语法:unalias name
例如:
unalias dir 删除别名 dir 的定义
unalias * 删除所有别名的设定

446 ☆echo

echo [-n] 字符串

echo 命令的功能是在显示器上显示一段文字，一般起到一个提示的作用。

其中选项 n 表示输出文字后不换行;字符串可以加引号，也可以不加引号。用 echo 命令输出加引号的字符串时，将字符串原样输出;用 echo 命令输出不加引号的字符串时，将字符串中的各个单词作为字符串输出，各字符串之间用一个空格分割。

echo something
这个命令回显所有接收到的内容

查看匹配
echo *sep*
echo *c[1-3]

echo $NAME
显示指定的环境变量 NAME 的设定值

echo $PS1

PS1='Red Hat Linux ->'

PS1='\h \! $'

echo $PRINTER 显示环境变量 PRINTER 的设定值

447 ☆set

语法: set 查看所有外壳变量的设定值

set -o
显示选项

$ set -o ignoreeof
$ <ctrl-d>
$ 用 "logout" 退出 shell
$ set +o ignoreeof
$ <ctrl-d>

set -o noclobber

$ echo "new contents" > trfile.txt
bash: trfile.txt：cannot overwrite existing file
重定向不能覆盖了

设定外壳变量
语法: set var=value

例如:
set term=vtl00 设定外壳变量 term 为 VT100 型终端

448 ☆unset

语法: unset var
例如:
unset PRINTER 删除外壳变量 PRINTER 的设定值

449 ☆setenv

语法: setenv
查看所有环境变量的设定值

语法: setenv NAME word

例如:
setenv PRINTER sp 设定环境变量 PRINTER 为 sp

450 ☆unsetenv

语法:unsetenv NAME
例如:
unsetenv PRINTER 删除环境变量 PRINTER 的设定值

451 ☆规则表达式

^ 与行首匹配
& 与行末尾匹配
. 与任一个字符匹配
* 将与 前一个字符的零或多个出现匹配
[ ] 与 [ ] 之内的所有字符匹配

/./ 将与包含至少一个字符的任何行匹配
/../ 将与包含至少两个字符的任何行匹配

/^#/ 将与以 '#' 开始的任何行匹配
/^$/ 将与所有空行匹配
/}^/ 将与以 '}'（无空格）结束的任何行匹配
/} *^/ 将与以 '}' 后面跟有 零或多个空格结束的任何行匹配
/[abc]/ 将与包含小写 'a'、'b' 或 'c' 的任何行匹配
/^[abc]/ 将与以 'a'、'b' 或 'c' 开始的任何行匹配

/[a-x]*/ 匹配零或多个全部为 'a'、'b'、'c'...'v'、'w'、'x' 的字符

[:alnum:] 字母数字 [a-z A-Z 0-9]
[:alpha:] 字母 [a-z A-Z]
[:blank:] 空格或制表键
[:cntrl:] 任何控制字符
[:digit:] 数字 [0-9]
[:graph:] 任何可视字符（无空格）
[:lower:] 小写 [a-z]
[:print:] 非控制字符
[:punct:] 标点字符
[:space:] 空格
[:upper:] 大写 [A-Z]
[:xdigit:] 十六进制数字 [0-9 a-f A-F]

452☆sed

sed 是一种流编辑器，所以，它可以对从如管道这样的标准输入接收的数据进行编辑。因此，无需将要编辑的数据存储在磁盘上的文件中。因为可以轻易将数据管道输出到 sed，所以，将 sed 用作强大的 shell 脚本中长而复杂的管道很容易。

sed 通过对输入数据执行任意数量用户指定的编辑操作（“命令”）来工作。sed 是基于行的，因此按顺序对每一行执行命令。然后，sed 将其结果写入标准输出 (stdout)，它不修改任何输入文件。

sed -e 'd' /etc/services
用一个编辑命令 'd' 调用 sed。sed 打开 /etc/services 文件，将一行读入其模式缓冲区，执行编辑命令（“删除行”），然后打印模式缓冲区（缓冲区已为空）。然后，它对后面的每一行重复这些步骤。这不会产生输出，因为 "d" 命令除去了模式缓冲区中的每一行！

'd' 命令不是简单地告诉 sed 一下子删除所有输入数据。相反，sed 逐行将 /etc/services 的每一行读入其称为模式缓冲区的内部缓冲区。一旦将一行读入模式缓冲区，它就执行 'd' 命令，然后打印模式缓冲区的内容（在本例中没有内容）。

要注意的事是括起 'd' 命令的单引号的用法。养成使用单引号来括起 sed 命令的习惯是个好注意，这样可以禁用 shell 扩展。

sed -e '1d' /etc/services | more
使用 sed 从输出流除去 /etc/services 文件第一行

sed -e '1,10d' /etc/services | more
删除输出的第 1 到 10 行
当用逗号将两个地址分开时，sed 将把后面的命令应用到从第一个地址开始、到第二个地址结束的范围。在本例中，将 'd' 命令应用到第 1 到 10 行（包括这两行）。所有其它行都被忽略。

sed -e '/^#/d' /etc/services | more
删除以 '#' 开始的行。
规则表达式地址总是由斜杠括起。它们指定一种模式，紧跟在规则表达式地址之后的命令将仅适用于正好与该特定模式匹配的行。

sed -e '/^#/p' /etc/services | more
p 是打印模式空间

sed -n -e '/BEGIN/,/END/p' /my/test/file | more
将打印从包含 "BEGIN" 的行开始，并且以包含 "END" 的行结束的文本块
如果没发现 "BEGIN"，那么将不打印数据。如果发现了 "BEGIN"，但是在这之后的所有行中都没发现 "END"，那么将打印所有后续行。发生这种情况是因为 sed 面向流的特性 -- 它不知道是否会出现 "END"。
sed 将与所有从匹配第一个规则表达式的第一行开始，到匹配第二个规则表达式的行结束（包括该行）的所有行匹配。

sed -e 's/foo/bar/' myfile.txt
将 myfile.txt 中每行第一次出现的 'foo'（如果有的话）用字符串 'bar' 替换，然后将该文件内容输出到标准输出。
请注意，这里是 每行第一次出现

sed -e 's/foo/bar/g' myfile.txt
替换每行中的 所有出现的 foo
最后一个斜杠之后附加的 'g' 选项告诉 sed 执行全局替换。

sed -e '1,10s/enchantment/entrapment/g' myfile2.txt
用短语 'entrapment' 替换所有出现的短语 'enchantment'，但是只在第一到第十行（包括这两行）

sed -e '/^$/,/^END/s/hills/mountains/g' myfile3.txt
用 'mountains' 替换 'hills'，但是，只从空行开始，到以三个字符 'END' 开始的行结束（包括这两行）的文本块上这样做

sed -e 's:/usr/local:/usr:g' mylist.txt

将把所有出现的 /usr/local 替换成 /usr
如果正在执行字符串替换，并且规则表达式或替换字符串中有许多斜杠，则可以通过在 's' 之后指定一个不同的字符来更改分隔符
当然了,如果需要在规则表达式中指定分隔符字符，可以在它前面加入反斜杠。

sed -e 's/<.*>//g' myfile.html
匹配从 '<' 开始、到 '>' 结束、并且在其中包含任意数量字符的短语。并将删除该短语（用空字符串替换）

se 在行中匹配规则表达式时，它总是在行中查找 最长的匹配
如果行是这样的:This is what I meant.
会出问题,变成 meant.

sed -e 's/<[^>]*>//g' myfile.html
这样可以解决
'[^>]' 指定“非 '>'”字符，其后的 '*' 完成该表达式以表示“零或多个非 '>' 字符”。

sed -e 's/.*/ralph said: &/' origmsg.txt
每一行前面加上短语 "ralph said: "
替换字符串中使用了 '&' 字符，该字符告诉 sed 插入整个匹配的规则表达式。

如这个文件
foo bar oni eeny meeny miny larry curly moe jimmy the weasel
把 "eeny meeny miny" 替换成 "Victor eeny-meeny Von miny" 这样的形式
首先要编写一个由空格分隔并与三个字符串匹配的规则表达式。
'.* .* .*'
将在其中每个感兴趣的区域两边插入带反斜杠的圆括号来定义区域
'\(.*\) \(.*\) \(.*\)'
最后
sed -e 's/\(.*\) \(.*\) \(.*\)/Victor \1-\2 Von \3/' myfile.txt

通过输入 '\x'（其中，x 是从 1 开始的区域号）来引用每个由圆括号定界的区域

输出成
Victor foo-bar Von oni Victor eeny-meeny Von miny Victor larry-curly Von moe Victor jimmy-the Von weasel

sed -n -e '=;p' myfile.txt
使用 '=' 命令和 'p' 命令，'=' 命令告诉 sed 打印行号，'p' 命令明确告诉 sed 打印该行（因为处于 '-n' 模式）。
首先将 '=' 命令应用到第 1 行，然后应用 'p' 命令。
输入多个命令可以在命令之间使用分号。

无论什么时候指定了两个或更多命令，都按顺序将每个命令应用到文件的每一行。

sed -n -e '=' -e 'p' myfile.txt
用两个 -e 选项来指定两个不同的命令,和上面的效果一样

sed -n -f mycommands.sed myfile.txt
是将命令放入一个单独的文件中。(更复杂的情况)

1,20{ s/[Ll]inux/GNU\/Linux/g s/samba/Samba/g s/posix/POSIX/g }
三个替换命令应用到第 1 行到第 20 行（包括这两行）
要对一个地址执行多个命令,用 '{ }' 字符将这些命令分组

1,/^END/{ s/[Ll]inux/GNU\/Linux/g s/samba/Samba/g s/posix/POSIX/g
p }
将把 '{ }' 之间的所有命令应用到从第 1 行开始，到以字母 "END" 开始的行结束（如果在源文件中没发现 "END"，则到文件结束）的所有行。

i\ This line will be inserted before each line
应用到每一行,在行首插入字串

i\ insert this line\ and this one\ and this one\ and, uh, this one too.
如果要在当前行之前插入多行，可以通过在前一行之后附加一个反斜杠来添加附加行

a\ insert this line after each line. Thanks! :)
将把一行或多行插入到模式空间中的当前行之后

c\ You're history, original line! Muhahaha!
将 替换 模式空间中的当前行,所有行都成替换的字符

sed -e 's/$/\r/' myunix.txt > mydos.txt
'$' 规则表达式将与行的末尾匹配，而 '\r' 告诉 sed 在其之前插入一个回车。在换行之前插入回车，立即，每一行就以 CR/LF 结束。
将 UNIX 风格的文本转换成 DOS/Windows 格式。您可能知道，基于 DOS/Windows 的文本文件在每一行末尾有一个 CR（回车）和 LF（换行），而 UNIX 文本只有一个换行。有时可能需要将某些 UNIX 文本移至 Windows 系统，该脚本将为您执行必需的格式转换

sed -e 's/.$//' mydos.txt > myunix.txt
将把 DOS/Windows 格式的文本转换成可信赖的 UNIX 格式
替代规则表达式与一行的最末字符匹配，而该字符恰好就是回车。我们用空字符替换它，从而将其从输出中彻底删除。

sed -e '1!G;h;$!d' forward.txt > backward.txt
实现和 tac 同样的功能
包含三个由分号隔开的单独 sed 命令：'1!G'、'h' 和 '$!d'。
如果第一个命令是 '1G'，则 'G' 命令将只应用第一行。然而，还有一个 '!' 字符 -- 该 '!' 字符 忽略该地址，即，'G' 命令将应用到除第一行之外的 所有行。
'$!d' 命令与之类似。如果命令是 '$d'，则将只把 'd' 命令应用到文件中的最后一行（'$' 地址是指定最后一行的简单方式）。然而，有了 '!' 之后，'$!d' 将把 'd' 命令应用到除最后一行之外的 所有行。
首先执行的命令是 'h'。该命令告诉 sed 将模式空间（保存正在处理的当前行的缓冲区）的内容复制到保留空间（临时缓冲区）。然后，执行 'd' 命令，该命令从模式空间中删除 "foo"，以便在对这一行执行完所有命令之后不打印它。

453☆awk

awk 的 GNU 版本（叫作 gawk）

awk '{ print }' /etc/passwd
依次对 /etc/passwd 中的每一行执行 print 命令。所有输出都发送到 stdout，所得到的结果与与执行 cat /etc/passwd 完全相同

在 awk 中，花括号用于将几块代码组合到一起，这一点类似于 C 语言。在代码块中只有一条 print 命令。在 awk 中，如果只出现 print 命令，那么将打印当前行的全部内容。

awk '{ print $0 }' /etc/passwd
$0 变量表示整个当前行，所以 print 和 print $0 的作用完全一样。

awk '{ print "" }' /etc/passwd
将 "" 字符串传递给 print 命令，它就会打印空白行。如果测试该脚本，将会发现对于 /etc/passwd 文件中的每一行，awk 都输出一个空白行。再次说明， awk 对输入文件中的每一行都执行这个脚本。

awk '{ print "hiya" }' /etc/passwd
运行这个脚本将在您的屏幕上写满 hiya

awk -F":" '{ print $1 }' /etc/passwd
打印出您的系统上所有用户帐户的列表
用 -F 选项来指定 ":" 作为字段分隔符
print $1 输入文件中每一行中出现的第一个字段

awk -F":" '{ print $1 $3 }' /etc/passwd
awk 打印出 /etc/passwd 文件的第一和第三个字段

awk -F":" '{ print $1 " " $3 }' /etc/passwd
连接 $1 、" " 和 $3 ，创建可读的输出
在这两个字段中插入空格

awk -F":" '{ print "username: " $1 "\t\tuid:" $3 }' /etc/passwd
插入一些文本标签

awk -f myscript.awk myfile.in
向 awk 传递 -f 选项，以向它提供此脚本文件
myscript.awk 文件中
```
BEGIN {
    FS=":"
}
{ print $1 }
```
字段分隔符在代码自身中指定（通过设置 FS 变量）

454 ☆awk 文件语法

对于每个输入行，awk 都会执行每个脚本代码块一次。然而，在许多编程情况中，可能需要在 awk 开始处理输入文件中的文本之前执行初始化代码。
awk 在开始处理输入文件之前会执行 BEGIN 块，因此它是初始化 FS（字段分隔符）变量、打印页眉或初始化其它在程序中以后会引用的全局变量的极佳位置。

awk 还提供了另一个特殊块，叫作 END 块。awk 在处理了输入文件中的所有行之后执行这个块。通常，END 块用于执行最终计算或打印应该出现在输出流结尾的摘要信息。

awk 允许使用规则表达式，根据规则表达式是否匹配当前行来选择执行独立代码块。

/foo/ { print }
只输出包含字符序列 foo 的那些行

/[0-9]+\.[0-9]*/ { print }
打印包含浮点数的行

将任意一种布尔表达式放在一个代码块之前，以控制何时执行某特定块。仅当对前面的布尔表达式求值为真时，awk 才执行代码块。

$1 == "fred" { print $3 }
输出将输出其第一个字段等于 fred 的所有行中的第三个字段。如果当前行的第一个字段不等于 fred ，awk 将继续处理文件而不对当前行执行 print 语句

awk 提供了完整的比较运算符集合，包括 "=="、"<"、">"、"<="、">=" 和 "!="。另外，awk 还提供了 "~" 和 "!~" 运算符，它们分别表示“匹配”和“不匹配”。它们的用法是在运算符左边指定变量，在右边指定规则表达式。

```
$5 ~ /root/ { print $3 }
```
如果某一行的第五个字段包含字符序列 root ，那么以下示例将只打印这一行中的第三个字段

条件语句

```
{
    if ( $5 ~ /root/ ) {
        print $3
    }
}
```

和上面的功能相同

更复杂的例子

```
{
    if ( $1 == "foo" ) {
        if ( $2 == "foo" ) {
            print "uno"
        } else {
            print "one"
        }
    } else if ($1 == "bar" ) {
        print "two"
    } else {
        print "three"
    }
}
```

```
! /matchme/ { print $1 $3 $4 }
```
这种语句可以转换为

```
{
    if ( $0 !~ /matchme/ ) {
        print $1 $3 $4
    }
}
```
输出 不 包含 matchme 字符序列的那些行。

awk 还允许使用布尔运算符 "||"（逻辑与）和 "&&"（逻辑或）

($1 == "foo") && ($2 == "bar") { print }

打印第一个字段等于 foo 且 第二个字段等于 bar 的那些行。

数值变量

```
BEGIN     { x=0 }
/^$/      { x=x+1 }
END       { print "I found " x " blank lines. :)" }
```

在 BEGIN 块中，将整数变量 x 初始化成零。
然后，awk 每次遇到空白行时，awk 将执行 x=x+1 语句，递增 x 。
处理完所有行之后，执行 END 块，awk 将打印出最终摘要，指出它找到的空白行数量。

字符串化变量
awk 变量“字符串化”是因为所有 awk 变量在内部都是按字符串形式存储的。同时，awk 变量是“简单的”，因为可以对它执行数学操作，且只要变量包含有效数字字符串，awk 会自动处理字符串到数字的转换步骤。

```
x="1.01"
# We just set x to contain the *string* "1.01"
x=x+1
# We just added one to a *string*
print x
# Incidentally, these are comments :)
```

awk 将输出:
2.01

```
{ print ($1^2)+1 }
```

对每个输入行的第一个字段乘方并加一

运算符包括前后加减（ i++ 、 --foo ）、加 / 减 / 乘 / 除赋值运算符（ a+=3 、 b*=2 、 c/=2.2 、 d-=6.2 ）,指数运算符 "^"、模（余数）运算符 "%"。不仅如此 -- 我们还有易于使用的模 / 指数赋值运算符（ a^=2 、 b%=4 ）。

awk 有它自己的特殊变量集合。其中一些允许调整 awk 的运行方式，而其它变量可以被读取以收集关于输入的有用信息。

FS 值并没有被限制为单一字符；可以通过指定任意长度的字符模式，将它设置成规则表达式。

```
FS="\t+"
```

处理由一个或多个 tab 分隔的字段
"+" 规则表达式字符，它表示“一个或多个前一字符”。

FS="[[:space:]+]"
字段由空格分隔（一个或多个空格或 tab）

FS="foo[0-9][0-9][0-9]"
记录由单词 "foo" 分隔，后面跟着三个数字

字段数量

NF 变量，也叫做“字段数量”变量。
awk 会自动将该变量设置成当前记录中的字段数量。
可以使用 NF 变量来只显示某些输入行

```
NF == 3 { print "this particular record has three fields: " $0 }
```
或者
```
{
    if ( NF > 2 ) {
        print $1 " " $2 ":" $3
    }
}
```
是一样的

记录号 (NR) 是另一个方便的变量。
它始终包含当前记录的编号（awk 将第一个记录算作记录号 1）。
迄今为止，我们已经处理了每一行包含一个记录的输入文件。
对于这些情况，NR 还会告诉您当前行号。

```
(NR < 10 ) || (NR > 100) { print "We are on record number 1-9 or 101+" }
```

```
{
    #skip header
    if ( NR > 10 ) {
        print "ok, now for the real information!"
    }
}
```

RS 变量告诉 awk 当前记录什么时候结束，新记录什么时候开始。

这样一个文件
Jimmy the Weasel
100 Pleasant Drive
San Francisco, CA 12345

Big Tony
200 Incognito Ave.
Suburbia, WA 67890
希望 awk 将每 3 行看作是一个独立的记录，而不是三个独立的记录。如果 awk 将地址的第一行看作是第一个字段 ($1)，街道地址看作是第二个字段 ($2)，城市、州和邮政编码看作是第三个字段 $3

```
BEGIN {
    FS="\n"
    RS=""
}
```

FS 设置成 "\n" 告诉 awk 每个字段都占据一行。通过将 RS 设置成 ""，还会告诉 awk 每个地址记录都由空白行分隔

```
BEGIN {
    FS="\n"
    RS=""
}
{
    print $1 ", " $2 ", " $3
}
```

将每个记录打印在一行上，用逗号分隔每个字段。
awk -f address.awk address.txt" 来执行这个脚本
这个只输出第一行

awk 变量 OFS
这是缺省情况下的输出结果，OFS 被设置成 " "，单个空格。

```
print "Hello", "there", "Jim!"
```

逗号并不是实际文字字符串的一部分。事实上，它们告诉 awk "Hello"、"there" 和 "Jim!" 是单独的字段，并且应该在每个字符串之间打印 OFS 变量。
输出
Hello there Jim!

```
BEGIN {
    FS="\n"
    RS=""
    OFS=", "
}
{
    print $1, $2, $3
}
```

OFS 来输出那些中间的 ", " 字符串

awk 还有一个特殊变量 ORS，全称是“输出记录分隔符”。通过设置缺省为换行 ("\n") 的 OFS，我们可以控制在 print 语句结尾自动打印的字符。缺省 ORS 值会使 awk 在新行中输出每个新的 print 语句。如果想使输出的间隔翻倍，可以将 ORS 设置成 "\n\n"。或者，如果想要用单个空格分隔记录（而不换行），将 ORS 设置成 ""。

将多行转换成用 tab 分隔的格式

将地址列表转换成每个记录一行，且用 tab 定界的格式，以便导入电子表格。使用稍加修改的 address.awk 之后，就可以清楚地看到这个程序只适合于三行的地址。如果 awk 遇到以下地址，将丢掉第四行，并且不打印该行：
Cousin Vinnie
Vinnie's Auto Shop
300 City Alley
Sosueme, OR 76543

考虑每个字段的记录数量，并依次打印每个记录。

```
BEGIN {
	FS="\n"
	RS=""
	ORS=""
}

{
		x=1
		while ( x<NF ) {
			print $x "\t"
			x++
		}
		print $NF "\n"
}
```

首先，将字段分隔符 FS 设置成 "\n"，将记录分隔符 RS 设置成 ""，这样 awk 可以象以前一样正确分析多行地址。
然后，将输出记录分隔符 ORS 设置成 ""，它将使 print 语句在每个调用结尾 不 输出新行。这意味着如果希望任何文本从新的一行开始，那么需要明确写入 print "\n" 。

在主代码块中，创建了一个变量 x 来存储正在处理的当前字段的编号。起初，它被设置成 1。然后，我们使用 while 循环（一种 awk 循环结构，等同于 C 语言中的 while 循环），对于所有记录（最后一个记录除外）重复打印记录和 tab 字符。最后，打印最后一个记录和换行；此外，由于将 ORS 设置成 ""，print 将不输出换行。程序输出如下，这正是我们所期望的：
Jimmy the Weasel	100 Pleasant Drive	San Francisco, CA 12345

Big Tony 200 Incognito Ave. Suburbia, WA 67890
Cousin Vinnie Vinnie's Auto Shop 300 City Alley Sosueme, OR 76543

循环结构

do...while 示例

```
{
      count=1
      do {
            print "I get printed at least once no matter what"
      } while ( count != 1 )
}
```

for 循环

```
for ( initial assignment; comparison; increment ) {
      code block
}
```

示例：

```
for ( x = 1; x <= 4; x++ ) {
      print "iteration",x
}
```

打印

```
iteration 1
iteration 2
iteration 3
iteration 4
```

break 和 continue

while 死循环

```
while (1) {
      print "forever and ever..."
}
```

while 死循环 1 永远代表是真，这个 while 循环将永远运行下去。

break 语句示例

```
x=1
while(1) {
      print "iteration",x
```

```
    if ( x == 10 ) {
        break
    }
    x++
}
```

break 语句用于“逃出”最深层的循环。"break" 使循环立即终止，并继续执行循环代码块后面的语句。

continue 语句补充了 break，其作用如下：

```
x=1
while (1) {
    if ( x == 4 ) {
        x++
        continue
    }
    print "iteration",x
    if ( x > 20 ) {
        break
    }
    x++
}
```

这段代码打印 "iteration 1" 到 "iteration 21"，"iteration 4" 除外。如果迭代等于 4，则增加 x 并调用 continue 语句，该语句立即使 awk 开始执行下一个循环迭代，而不执行代码块的其余部分。如同 break 一样，continue 语句适合各种 awk 迭代循环。在 for 循环主体中使用时，continue 将使循环控制变量自动增加。以下是一个等价循环：

```
for ( x=1; x<=21; x++ ) {
    if ( x == 4 ) {
        continue
    }
    print "iteration",x
}
```

在 while 循环中时，在调用 continue 之前没有必要增加 x ，因为 for 循环会自动增加 x 。

数组

在 awk 中，数组下标通常从 1 开始，而不是 0

```
myarray[1]="jim"
myarray[2]=456
```

awk 遇到第一个赋值语句时，它将创建 myarray ，并将元素 myarray[1] 设置成 "jim"。执行了第二个赋值语句后，数组就有两个元素了。

数组迭代
定义之后，awk 有一个便利的机制来迭代数组元素，如下所示：

```
for ( x in myarray ) {
      print myarray[x]
}
```

打印数组 myarray 中的每一个元素。当对于 for 使用这种特殊的 "in" 形式时，awk 将 myarray 的每个现有下标依次赋值给 x （循环控制变量），每次赋值以后都循环一次循环代码。虽然这是一个非常方便的 awk 功能，但它有一个缺点 -- 当 awk 在数组下标之间轮转时，它不会依照任何特定的顺序。那就意味着我们不能知道以上代码的输出是:

```
jim
456
```

还是

```
456
jim
```

数组下标字符串化

```
a="1"
b="2"
c=a+b+3
```

执行了这段代码后， c 等于 6 。由于 awk 是“字符串化”的，添加字符串 "1" 和 "2" 在功能上并不比添加数字 1 和 2 难。这两种情况下，awk 都可以成功执行运算。awk 的“字符串化”性质非常可爱 -- 您可能想要知道如果使用数组的字符串下标会发生什么情况。

```
myarr["1"]="Mr. Whipple"
print myarr["1"]
```

这段代码将打印 "Mr. Whipple"。但如果去掉第二个 "1" 下标中的引号

```
myarr["1"]="Mr. Whipple"
print myarr[1]
```

awk 将 myarr["1"] 和 myarr[1] 看作数组的两个独立元素，还是它们是指同一个元素？答案是它们指的是同一个元素，awk 将打印 "Mr. Whipple"，如同第一个代码片断一样。虽然看上去可能有点怪，但 awk 在幕后却一直使用数组的字符串下标！

这段代码不仅不会产生错误，而且它的功能与前面的示例完全相同，也将打印 "Mr. Whipple"！可以看到，awk 并没有限制我们使用纯整数下标；如果我们愿意，可以使用字符串下标，而且不会产生任何问题。只要我们使用非整数数组下标，如 myarr["name"] ，那么我们就在使用 关联数组 。从技术上讲，如果我们使用字符串下标，awk 的后台操作并没有什么不同（因为即便使用“整数”下标，awk 还是会将它看作是字符串）。但是，应该将它们称作 关联数组 -- 它听起来很酷，而且会给您的上司留下印象。字符串化下标是我们的小秘密。;)

数组工具

可以使用字符串下标，而且不需要连续的数字序列下标（例如，可以定义 myarr[1] 和 myarr[1000] ，但不定义其它所有元素）。虽然这些都很有用，但在某些情况下，会产生混淆。幸好，awk 提供了一些实用功能有助于使数组变得更易于管理。

首先，可以删除数组元素。如果想要删除数组 fooarray 的元素 1 ，输入：
delete fooarray[1]

如果想要查看是否存在某个特定数组元素，可以使用特殊的 "in" 布尔运算符
```
if ( 1 in fooarray ) {
	print "Ayep!  It's there."
} else {
	print "Nope!  Can't find it."
}
```

格式化输出

printf() 会将格式化字符串打印到 stdout，而 sprintf() 则返回可以赋值给变量的格式化字符串。

```
x=1
b="foo"
printf("%s got a %d on the last test\n","Jim",83)
myout=("%s-%d",b,x)
print myout
```

此代码将打印：
```
Jim got a 83 on the last test
foo-1
```

字符串函数

```
mystring="How are you doing today?"
print mystring[3]
```

将会接收到一个错误，如下所示：
awk: string.gawk:59: fatal: attempt to use scalar as array
不能象在其它语言（如 C、C++ 和 Python）中那样将字符串看作是字符数组

length() 函数，它返回字符串的长度

```
print length(mystring)
```
此代码将打印值：

index，它将返回子字符串在另一个字符串中出现的位置，如果没有找到该字符串则返回 0

print index(mystring,"you")
awk 会打印：
9

tolower() 和 toupper()。这两个函数将返回字符串并且将所有字符分别转换成小写或大写
tolower() 和 toupper() 返回新的字符串，不会修改原来的字符串。

print tolower(mystring)
print toupper(mystring)
print mystring

将产生以下输出：
how are you doing today?
HOW ARE YOU DOING TODAY?
How are you doing today?

substr() 的调用方法：
mysub=substr(mystring,startpos,maxlen)
mystring 应该是要从中抽取子串的字符串变量或文字字符串。startpos 应该设置成起始字符位置，maxlen 应该包含要抽取的字符串的最大长度。请注意，我说的是 最大长度 ；如果 length(mystring) 比 startpos+maxlen 短，那么得到的结果就会被截断。substr() 不会修改原始字符串，而是返回子串。

print substr(mystring,9,3)
awk 将打印：
you
如果您通常用于编程的语言使用数组下标访问部分字符串（以及不使用这种语言的人），请记住 substr() 是 awk 代替方法。需要使用它来抽取单个字符和子串；因为 awk 是基于字符串的语言，所以会经常用到它

match() 与 index() 非常相似，它与 index() 的区别在于它并不搜索子串，它搜索的是规则表达式。match() 函数将返回匹配的起始位置，如果没有找到匹配，则返回 0。此外，match() 还将设置两个变量，叫作 RSTART 和 RLENGTH。RSTART 包含返回值（第一个匹配的位置），RLENGTH 指定它占据的字符跨度（如果没有找到匹配，则返回 -1）。通过使用 RSTART、RLENGTH、substr() 和一个小循环，可以轻松地迭代字符串中的每个匹配。

print match(mystring,/you/), RSTART, RLENGTH
打印：

字符串替换

sub(regexp,replstring,mystring)
在 mystring 中匹配 regexp 的第一个字符序列，并且用 replstring 替换该序列。

sub() 和 gsub() 用相同的自变量;唯一的区别是 sub() 将替换第一个 regexp 匹配(如果有的话)，gsub() 将执行全局替换，换出字符串中的所有匹配。

```
sub(/o/,"O",mystring)
print mystring
mystring="How are you doing today?"
gsub(/o/,"O",mystring)
print mystring
```
输出：
```
HOw are you doing today?
HOw are yOu dOing tOday?
```

split() 的任务是“切开”字符串，并将各部分放到使用整数下标的数组中。

```
numelements=split("Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec",mymonths,",")
```
用 split() 时，第一个自变量包含要切开文字字符串或字符串变量。在第二个自变量中，应该指定 split() 将填入片段部分的数组名称。在第三个元素中，指定用于切开字符串的分隔符。
split() 返回时，它将返回分割的字符串元素的数量。split() 将每一个片段赋值给下标从 1 开始的数组

```
print mymonths[1],mymonths[numelements]
```
打印：
```
Jan Dec
```

调用 length()、sub() 或 gsub() 时，可以去掉最后一个自变量，这样 awk 将对 $0（整个当前行）应用函数调用。

```
{
    print length()
}
```
要打印文件中每一行的长度

455 ☆Xorg

Xorg 成为默认的 X11 系统。Xorg 是由 X.Org 组织发布的，根据 X11R6.7 所制作的 X11 服务器。X11R6.7 是基于 XFree86 4.4RC2 的代码和 X11R6.6。

456 ☆X

/etc/X11/xorg.conf
配置文件

X -configure
配置 X,先备份 xorg.conf

457 ☆安装 x11/xorg

与 XFree86 的差异只有几个配置方式作了改变。比如程序 XFree86(1) 变成 Xorg(1)、xf86cfg(1) 变成 xorgcfg(1)、xf86config(1) 变成 xorgconfig(1)。配置文件 XF86Config 变成 xorg.conf。日志文件 XFree86.0.log 变成 Xorg.0.log。

458 ☆xorg.conf

/etc/X11/xorg.conf
首先 xorg.conf 只能由 root 用户打开，所以先升级到 root 用户：
输入 su 命令，然后输入 root 的密码，这样就有 root 权限了
然后用 vi 或 vim 命令编辑
vim xorg.conf

对 xorg.conf 的修改要慎重，修改前先备份

```
# XFree86 4 configuration created by pyxf86config

Section "ServerLayout"
        Identifier     "Default Layout"
        Screen      0  "Screen0" 0 0
        InputDevice    "Mouse0" "CorePointer"
        InputDevice    "Keyboard0" "CoreKeyboard"
EndSection

Section "Files"

# RgbPath is the location of the RGB database.  Note, this is the name of the
# file minus the extension (like ".txt" or ".db").  There is normally
```

```
# no need to change the default.
# Multiple FontPath entries are allowed (they are concatenated together)
# By default, Red Hat 6.0 and later now use a font server independent of
# the X server to render fonts.
        RgbPath      "/usr/X11R6/lib/X11/rgb"
        FontPath     "unix/:7100"
EndSection

Section "Module"
        Load  "dbe"
        Load  "extmod"
        Load  "fbdevhw"
        Load  "glx"
        Load  "record"
        Load  "freetype"
        Load  "type1"
        Load  "dri"
EndSection

Section "InputDevice"

# Specify which keyboard LEDs can be user-controlled (eg, with xset(1))
#       Option    "Xleds"         "1 2 3"
# To disable the XKEYBOARD extension, uncomment XkbDisable.
#       Option    "XkbDisable"
# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults).  For example, for a non-U.S.
# keyboard, you will probably want to use:
#       Option    "XkbModel"      "pc102"
# If you have a US Microsoft Natural keyboard, you can use:
#       Option    "XkbModel"      "microsoft"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
#       Option    "XkbLayout"     "de"
# or:
#       Option    "XkbLayout"     "de"
#       Option    "XkbVariant"    "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
#       Option    "XkbOptions"    "ctrl:swapcaps"
```

```
# Or if you just want both to be control, use:
#       Option    "XkbOptions"       "ctrl:nocaps"
#
        Identifier    "Keyboard0"
        Driver          "kbd"
        Option          "XkbModel" "pc105"
        Option          "XkbLayout" "us"
EndSection

Section "InputDevice"
        Identifier    "Mouse0"
        Driver          "mouse"
        Option          "Protocol" "IMPS/2"
        Option          "Device" "/dev/input/mice"
        Option          "ZAxisMapping" "4 5"
        Option          "Emulate3Buttons" "yes"
EndSection

Section "Monitor"
        Identifier     "Monitor0"
        VendorName     "Monitor Vendor"
        ModelName      "Unknown monitor"
        HorizSync       31.5 - 37.9
        VertRefresh    50.0 - 70.0
        Option          "dpms"
EndSection

Section "Device"
        Identifier    "Videocard0"
        Driver          "vmware"
        VendorName    "Videocard vendor"
        BoardName     "VMWare"
EndSection

Section "Screen"
        Identifier "Screen0"
        Device         "Videocard0"
        Monitor       "Monitor0"
        DefaultDepth         24
        SubSection "Display"
                Viewport     0 0
                Depth         16
```

```
                Modes       "800x600" "640x480"
        EndSubSection
        SubSection "Display"
                Viewport     0 0
                Depth        24
                Modes       "800x600" "640x480"
        EndSubSection
EndSection

Section "DRI"
        Group            0
        Mode             0666
EndSection
```

459 ☆Xorg -configure

自动创建 xorg.conf
Xorg -configure

当 Xorg 检测完你的硬件后，请务必注意屏幕上的最后一行显示。如果它告诉你某个地方检测失败，那么你就不得不自己手工编辑一个 xorg.conf 文件;如果没有，那么它会告诉你 /root/xorg.conf.new 已经创建并可以供你测试了。那么，就让我们测试一下吧
Xorg -config /root/xorg.conf.new

460 ☆xorgconfig

半自动创建 xorg.conf
Xorg 提供了一个名叫 xorgconfig 的工具，它将询问你有关系统的各种信息（图形适配器、键盘……）。它将根据你输入的信息来创建 xorg.conf 文件。
xorgconfig

461 ☆startx

现在运行 startx 来启动你的 X 服务器，它将使用你刚才复制过来的文件作为配置文件。要结束 X 会话进程，你只需要在活动的 xterms 中输入 exit 或者直接按下 Ctrl-Alt-Backspace 组合键即可，也许后者的方法显得不太优雅－－有时你并不想这样做。当然，这对你的系统并无大碍

启动 X

```
userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/etc/X11/xinit/xinitrc
```

```
sysserverrc=/etc/X11/xinit/xserverrc
defaultclient=/usr/X11R6/bin/xterm
defaultserver=/usr/X11R6/bin/X
defaultclientargs=""
defaultserverargs=""
clientargs=""
serverargs=""
```

462 ☆设置分辨率

如果你觉得屏幕分辨率有点问题，你需要检查配置文件中的两个段落。首先看看 Screen 段落，它列出了 X 服务器可以运行的所有分辨率。在默认情况下，这个段落往往不存在任何分辨率。如果是这种情况，Xorg 将根据 Monitor 段落中的信息来给出一个大概的分辨率。

Xorg 检查 Monitor 段落中的 HorizSync 和 VertRefresh 设置后会计算出一个有效的分辨率。眼下，先把这些东西放到一边来更改分辨率。在接下来的/etc/X11/xorg.conf 的例子中，我们将加入 Modes 和 DefaultDepth 使 X 服务器把 24 位色深和 1024x768 分辨率作为默认值启动。

修改/etc/X11/xorg.conf 中的 Screen 段落

```
Section "Screen"
   Identifier    "Default Screen"
   Device        "S3 Inc. ProSavage KN133 [Twister K]"
   Monitor      "Generic Monitor"
   DefaultDepth    24
   # 为了便于寻找和阅读请跳过一些内容
   SubSection "Display"
      Depth      24
      Modes      "1024x768"
   EndSubSection
EndSection
```

运行 X (startx)，看它是否启用了你所需要的分辨率

463 ☆配置键盘

要设置 X 以使用国际键盘布局，请找到与键盘设置相关的 InputDevice 段落，并且加入 XkbLayout 选项来指明你想使用的键盘布局。比如，接下来我们将为你说明如何应用比利时的键盘布局。你只需要将国家代码替换为你自己的即可。

更改键盘布局

```
Section "InputDevice"
   Identifier    "Generic Keyboard"
   Driver       "keyboard"
   Option       "CoreKeyboard"
   Option       "XkbRules"    "xorg"
   Option       "XkbModel"    "pc105"
   Option       "XkbLayout" "be"
EndSection
```

464 ☆配置鼠标

如果你的鼠标没有正常工作，你应该首先检查你的鼠标是否已经被系统内核检测到。PS/2 鼠标应该是作为/dev/psaux 出现的。而其他鼠标（比如 USB）则以/dev/input (或者 /dev/input/mice)出现。在任何一种情况下，你可以移动一下你的鼠标，然后检查这些文件的输出以确认这些设备文件是否已经指向你的鼠标。要结束，请按 Ctrl-C。

检查设备文件
cat /dev/input
(不要忘记按 Ctrl-C 来结束这项工作)

如果你的鼠标未被检测到，请检查必要的模块是否已经被载入。

如果你的鼠标已经被检测到，那么就在 InputDevice 段落中合适的位置填入你的设备。在接下来的例子中，你可以看到我们将设置其它两个选项：Protocol（它列出了可以使用的鼠标协议－－大多数用户使用的是 PS/2 或者 IMPS/2）以及 ZAxisMapping（对鼠标滚轮的支持（如果可用的话））。

在 Xorg 中改变鼠标设置

```
Section "InputDevice"
   Identifier    "TouchPad Mouse"
   Driver       "mouse"
   Option       "CorePointer"
   Option       "Device"       "/dev/psaux"
   Option       "Protocol"       "IMPS/2"
   Option       "ZAxisMapping"       "4 5"
EndSection
```

运行 startx 然后期待

465 ☆ddcprobe

显示器的刷新率和各种指标

466 ☆switchdesk

切换桌面到 KDE, GNOME, XFCE, FVWM or WindowMaker
switchdesk KDE

467 ☆prefdm

/etc/X11/prefdm

preferred=
设置首选的显示管理器
preferred=gdm
preferred=kdm
preferred=xdm

468 ☆gdmsetup

GNOME 桌面管理器

469 ☆Xorg.0.log

/var/log/Xorg.0.log
日志文件

EE 是错误信息
WW 是警告信息
II 是信息性信息

470 ☆setup

可以配置很多服务

471 ☆网络接口(interface)

网络接口(interface)是网络硬件设备在操作系统中的表示方法
比如网卡在 Linux 操作系统中用 ethX,是由 0 开始的正整数，比如 eth0、eth1..... ethX。
而普通猫和 ADSL 的接口是 pppX，比如 ppp0 等;

472 ☆ifconfig

ifconfig
如果不接任何参数，就会输出当前网络接口的情况;

ifconfig -a
如果我们想知道主机所有网络接口的情况，请用下面的命令;

ifconfig eth0
如果我们想查看某个端口，比如我们想查看 eth0 的状态，

ifconfig eth0 irq 9

ifconfig eth0 io_addr 0x003

ifconfig 配置网络端口的方法：
ifconfig 网络端口 IP 地址 hw <HW> MAC 地址 netmask 掩码地址 broadcast 广播地址 [up/down]

ifconfig eth0 down
ifconfig eth0 192.168.1.99 broadcast 192.169.152.12855 netmask 255.255.255.0
ifconfig eth0 up
ifconfig eth0

设置网络 IP 地址的同时，设置网卡的物理地址（MAC 地址）;
ifconfig eth1 192.169.152.128 hw ether 00:11:00:00:11:11 netmask 255.255.255.0 broadcast 192.169.152.255 up
或
ifconfig eth1 hw ether 00:11:00:00:11:22
ifconfig eth1 192.169.152.12852 netmask 255.255.255.0 broadcast 192.169.152.255 up
其中 hw 后面所接的是网络接口类型， ether 表示乙太网， 同时也支持 ax25 、ARCnet、netrom 等，详情请查看 man ifconfig ;

ifconfig 来配置虚拟网络接口;

虚拟网络接口指的是为一个网络接口指定多个 IP 地址，虚拟接口是这样的 eth0:0 、 eth0:1、 eth0:2 ... .. eth1N。

ifconfig eth1:0 192.169.152.12851 hw ether 00:11:00:00:11:33 netmask 255.255.255.0 broadcast 192.169.152.12855 up
或
[root@localhost ~]# ifconfig eth1:0 hw ether 00:11:00:00:11:33
[root@localhost ~]# ifconfig eth1:0 192.169.152.128 netmask 255.255.255.0 broadcast 192.169.152.12855 up

ifconfig 来激活和终止网络接口的连接;
激活和终止网络接口的用 ifconfig 命令，后面接网络接口，然后加上 down 或 up 参数，就可以禁止或激活相应的网络接口了。当然也可以用专用工具 ifup 和 ifdown 工具;

ifconfig eth0 down
ifconfig eth0 up
ifup eth0
ifdown eth0

注意：对 DHCP 自动分配的 IP，还得由各个发行版自带的网络工具来激活;当然得安装 dhcp 客户端;这个您我们应该明白;
/etc/init.d/network start

473☆ifdown

ifdown eth0
关闭网络接口

474☆ifup

ifup eth0
启动网络接口

475☆ifcfg-ethN

在 RedHat 中，系统网络设备的配置文件保存在"/etc/sysconfig/network-scripts"目录下，ifcfg-eth0 包含第一块网卡的配置信息，ifcfg-eth1 包含第二块网卡的配置信息等，若希望手工修改网络地址或在新的接口上增加新的网络界面，可以通过修改对应的文件（ifcfg-ethN）或创建新的文件来实现。

NETWORK=addr addr 表示网络地址
BROADCAST=addr addr 表示广播地址
NETWORKING=yes 一定要保证
USERCTL=yes/no 是否允许非 root 用户控制该设备

DEVICE=eth0 注：网络接口
ONBOOT=yes 注：开机引导时激活
BOOTPROTO=static 注：采用静态 IP 地址;dhcp,dialup,bootp 也可
none：无须启动协议
bootp：使用 bootp 协议
dhcp：使用 dhcp 协议

IPADDR=192.168.152.128 注：IP 地址
NETMASK=255.255.255.0 注：网络掩码;
GATEWAY=192.169.152.128 注：网关;
HOSTNAME=panda 注：指定主机名;
DOMAIN=panda.com 注：指定域名;
HWADDR=00:11:22:33:44:aa 注：指定网卡硬件地址 (MAC 地址）， 也可以省略，不过这在这里来更改 MAC 地址一般是不能生效的。还是通过前面所说的 ifconfig 的办法来更改吧;

476☆network

/etc/sysconfig/network
NETWORKING=yes
HOSTNAME=panda

这里更改主机名

477☆hosts

在引入 DNS(Domain Name System，域名系统)之前，是将容易记忆的域名映射到 IP 地址并将它保存在一个共享的静态文件 hosts 中，再由它来实现网络中域名的管理。最初 Internet 非常小，仅使用这个集中管理的文件就可以通过 FTP 为连入 Internet 的站点和主机提供域名的发布和下载。每个 Internet 站点将定期地更新其主机文件的副本，并发布主机文件的更新版本来反映网络的变化。开发 DNS 标准为主机文件提供了可供选择的方案。通过 DNS 将存储在数据库中的域名数据分布在不同的服务器上，就可以减轻对任何一台服务器的负载，并且提供了以区域为基础的对域名系统的分布式管理能力。

hosts 文件是 Linux 系统中一个负责 IP 地址与域名快速解析的文件，以 ASCII 格式保存在“/etc" 目录下，文件名为“hosts”。

hosts 文件包含了 IP 地址和主机名之间的映射，还包括主机名的别名。
在没有域名服务器的情况下，系统上的所有网络程序都通过查询该文件来解析对应于某个主机名的 IP 地址，否则就需要使用 DNS 服务程序来解决。
通常可以将常用的域名和 IP 地址映射加入到 hosts 文件中，实现快速方便的访问。
当机器启动时，在可以查询 DNS 以前，机器需要查询一些主机名到 IP 地址的匹配。这些匹配信息存放在/etc/hosts 文件中。在没有域名服务器情况下，系统上的所有网络程序都通过查询该文件来解析对应于某个主机名的 IP 地址。

/etc/hosts 文件
hosts 文件的格式如下：
IP 地址　　主机名/域名

192.168.16.177 center.panda.com

192.168.152.128 news.panda.com
192.168.152.129 mail.panda.com

最左边一列是主机 IP 信息，中间一列是主机名。任何后面的列都是该主机的别名。一旦配置完机器的网络配置文件，应该重新启动网络以使修改生效。使用下面的命令来重新启动网络：/etc/rc.d/init.d/network restart。

478☆resolv.conf

/etc/resolv.conf 文件

该文件是由域名解析器（resolver，一个根据主机名解析 IP 地址的库）使用的配置文件，示例如下：
search panda.com
nameserver 192.168.152.128
nameserver 192.168.152.129

“search panda.com ”定义了如果出现简单的不符合完整域名的主机名称时默认添加的缺省域。表示当提供了一个不包括完全域名的主机名时，在该主机名后添加 panda.com 的后缀;
“nameserver”表示解析域名时使用该地址指定的主机为域名服务器。其中域名服务器是按照文件中出现的顺序来查询的。指定了将主机 192.168.152.128 来作为 DNS 查询的解析器。最多 3 个 nameserver 行 <DNS 会从上往下尝试解析，直到成功>

您也应该注意到当您的系统重新启动的时候或者重新设定您的网络的时候您的/etc/resolve.conf 将会被改写

479☆host.conf

/etc/host.conf 文件
定义搜索顺序

该文件指定如何解析主机名。Linux 通过解析器库来获得主机名对应的 IP 地址。下面是一个“/etc/host.conf”的示例：
order bind,hosts
multi on
ospoof on

“order bind,hosts”指定主机名查询顺序，这里规定先使用 DNS 来解析域名，然后再查询“/etc/hosts”文件(也可以相反)。

“multi on”指定是否“/etc/hosts”文件中指定的主机可以有多个地址，拥有多个 IP 地址的主机一般称为多穴主机。

“nospoof on”指不允许对该服务器进行 IP 地址欺骗。IP 欺骗是一种攻击系统安全的手段，通过把 IP 地址伪装成别的计算机，来取得其它计算机的信任。

480 ☆system-config-network

网络配置工具(较全的)

481 ☆netconfig

在 setup 中调用的
netconfig 的用法如下：

netconfig --help 注：帮助;
-d, --device=STRING Network device （指定网络设备）
--bootproto=(dhcp|bootp|none) Boot protocol to use
--gateway=STRING Network gateway（指定网关）
--ip=STRING IP address（指定 IP 地址）
--nameserver=STRING Nameserver（指定 DNS 客户端）
--netmask=STRING Netmask（指定网络掩码）
--hostname=STRING Hostname（指定主机名）
--domain=STRING Domain name（指定域名）
--nodns No DNS lookups （没有 DNS 查询）
--hwaddr=STRING Ethernet hardware address （指定网卡的物理地址）
--description=STRING Description of the device （描述性文字）
Help options: (帮助选项）
-?, --help Show this help message
--usage Display brief usage message

netconfig -d eth0 注：配置 eth0

netconfig -d eth0 --bootproto=dhcp
设置网卡的 DHCP 模式自动获得 IP

netconfig -d eth0 --ip=192.168.1.33 --netmask=255.255.255.0 --gateway=192.169.152.129
手动设置网卡的 IP 等

482 ☆adsl-setup

配置 ADSL

[root@localhost ~]# adsl-setup 注：配置 pppoe 拔号，请文档下面，都差不多;

[root@localhost ~]# adsl-start 注：启动拔号;
[root@localhost ~]# adsl-stop 注：断开连接;

483☆wvdial

配置拨号猫

wvdial 工具是文本的，几乎在各大发行版都有。wvdial 的配置文件是/etc/wvdial.conf 。
如果您的猫已经驱动好了，运行一下 wvdialconf 命令就生成了/etc/wvdial.conf 了 。当然您得查看一下它的内容;
普通猫的拔号工具
[root@localhost ~]# wvdialconf
[root@localhost ~]# more /etc/wvdial.conf

484☆miitool

mii-tool - view, manipulate media-independent interface status
mii-tool 是查看，管理介质的网络接口的状态

有时网卡需要配置协商方式 ，比如 10/100/1000M 的网卡半双工、全双工、自动协商的配置 。但大多数的网络设备是不用我们来修改协商，因为大多数网络设置接入的时候，都采用自动协商来解决相互通信的问题。不过自动协商也不是万能的，有时也会出现错误，比如丢包率比较高，这时就要我们来指定网卡的协商方式 。

mii-tool
就是能指定网卡的协商方式。

参数:
[root@localhost ~]# mii-tool --help
usage: mii-tool [-VvRrwl] [-A media,... | -F media] [interface ...]
-V, --version display version information
-v, --verbose more verbose output 注：显示网络接口的信息;
-R, --reset reset MII to poweron state 注：重设 MII 到开启状态;
-r, --restart restart autonegotiation 注：重启自动协商模式;
-w, --watch monitor for link status changes 注：查看网络接口连接的状态变化;
-l, --log with -w, write events to syslog 注：写入事件到系统日志;
-A, --advertise=media,... advertise only specified media 注：指令特定的网络接口;
-F, --force=media force specified media technology 注：更改网络接口协商方式;
media: 100baseT4, 100baseTx-FD, 100baseTx-HD, 10baseT-FD, 10baseT-HD,
(to advertise both HD and FD) 100baseTx, 10baseT

mii-tool 在更改网络设备通信协商方式的方法比较简单，用 -v 参数来查看网络接口的状态;看下面的例子;

查看网络接口的协商状态;
[root@localhost ~]# mii-tool -v eth0
eth0: negotiated 100baseTx-FD, link ok
  product info: vendor 00:00:00, model 0 rev 0
  basic mode: autonegotiation enabled
  basic status: autonegotiation complete, link ok
  capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control

注：上面的例子，我们可以看得到是自动协商。

更改网络接口协商方式;
更改网络接口的协商方式，我们要用到-F 选项，后面可以接 100baseT4, 100baseTx-FD, 100baseTx-HD, 10baseT-FD, 10baseT-HD 等参数;

如果我们想把网络接口 eth0 改为 1000Mb/s 全双工的模式应该怎么办呢？
[root@localhost ~]# mii-tool -F 100baseTx-FD

[root@localhost ~]#mii-tool -v eth0
eth0: 100 Mbit, full duplex, link ok
  product info: vendor 00:00:00, model 0 rev 0
  basic mode: 100 Mbit, full duplex
  basic status: link ok
  capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

注：是不是已经改过来了？

当然，我们也一样用 ethtool 工具来更改，比如执行下面的命令;
[root@localhost ~]# ethtool -s eth0 speed 100 duplex full

485☆ethtool

ethtool - Display or change ethernet card settings
ethtool 是用来显示和更改网卡设置的工具;

ethtool 显示网络端口设置功能;

这个功能比较好办。就是 ethtool 后面直接接网絽接口就行;比如下面的例子;

```
[root@localhost ~]# ethtool eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes: 10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes: 10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Advertised auto-negotiation: No 注：自动协商关闭
        Speed: 100Mb/s 注：速度 100Mb
        Duplex: Full 注：全双工
        Port: MII
        PHYAD: 32
        Transceiver: internal
        Auto-negotiation: off
        Supports Wake-on: pumbg
        Wake-on: d
        Current message level: 0x00000007 (7)
        Link detected: yes 注：eth0 已经激活;
```

在 ethtool 的-h 帮助中我们查看到有这样的帮助信息;

```
ethtool -s DEVNAME \
                [ speed 10|100|1000 ] \
                [ duplex half|full ] \
                [ port tp|aui|bnc|mii|fibre ] \
                [ autoneg on|off ] \
```

把网卡 eth0 速度改为 10Mb/s，采用半双工;

```
[root@cuc03 panda]# ethtool -s eth1 speed 10 duplex half
[root@cuc03 panda]# ethtool eth1
Settings for eth1:
        Supported ports: [ TP MII ]
        Supported link modes:    10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Advertised auto-negotiation: No
        Speed: 10Mb/s 注：速度 10M/s
        Duplex: Half  注：半双工
```

Port: MII
PHYAD: 32
Transceiver: internal
Auto-negotiation: off
Supports Wake-on: pumbg
Wake-on: d
Current message level: 0x00000007 (7)
Link detected: no 注：eth1 没有激活;

把网卡 eth0 速度改为 100Mb/s，采用全双工;
[root@cuc03 panda]# ethtool -s eth1 speed 100 duplex full
[root@cuc03 panda]# ethtool eth1
Settings for eth1:
Supported ports: [TP MII]
Supported link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
Advertised auto-negotiation: No
Speed: 100Mb/s 注：速度 100M/s
Duplex: Full 注：全双工
Port: MII
PHYAD: 32
Transceiver: internal
Auto-negotiation: off
Supports Wake-on: pumbg
Wake-on: d
Current message level: 0x00000007 (7)
Link detected: no 注：eth1 网卡没有激活;

486☆netstat

netstat 命令的功能是显示网络连接、路由表和网络接口信息，可以让用户得知目前都有哪些网络连接正在运作。

语法：
netstat [选项]

命令中各选项的含义如下：
-a 显示所有 socket，包括正在监听的和未监听的 socket。
-r 显示核心路由表，格式同"route -e"。

-c 每隔 1 秒就重新显示一遍，直到用户中断它。
-i 显示所有网络接口的信息，格式同"ifconfig -e"。
-n 以网络 IP 地址代替名称，显示出网络连接情形。
-t 显示 TCP 协议的连接情况。
-u 显示 UDP 协议的连接情况。
-v 显示正在进行的工作。
-p 显示 socket 的 PID 和程序名

[root@panda ~]#netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags Types State I-Node Path
Unix 5 [ ] DGRAM 460 /dev/log
Unix 0 [ ] STREAM CONNECTED 173 @00000014
Unix 0 [ ] DGRAM 662
Unix 0 [ ] DGRAM 631
Unix 0 [ ] DGRAM 544
Unix 0 [ ] DGRAM 484
Unix 0 [ ] DGRAM 470

从整体上看，netstat 的输出结果可以分为两个部分：第一部分：是 Active Internet connections，称为有源 TCP 连接，在上面的输出结果中，这一部分没有内容，表示暂时还没有 TCP 连接。
第二部分：是 Active UNIX domain sockets，称为有源 Unix 域套接口。输出结果显示的是 Unix 域套接口的连接情况：

- Proto 显示连接使用的协议。
- RefCnt 表示连接到本套接口上的进程号。
- Types 显示套接口的类型。
- State 显示套接口当前的状态。
- Path 表示连接到套接口的其它进程使用的路径名。

返回的 Flag 的意义
Flag Description
G The route uses a gateway.
U The network adapter (Iface) is up.
H Only a single host can be reached via this route.D This entry was created by an ICMP redirect message.
M This entry was modified by an ICMP redirect message.

netstat -a
查看所有套接字的状态，这在您调试网络程序的时候是非常有用的。

netstat -r
将显示路由表的内容，一般还要同时指定"-n"选项，这样可以得到数字格式的地址，也可显示默认路由器的 IP 地址。

netstat -i
则将显示所有的网络接口信息。使用 netstat 还可以获得当前的网络状态以及网络的拓扑结构，这在实际中是非常有用的。

netstat -tualp
(TCP, UDP, ALL, Listen, PID)

netstat -taupe
(TCP, UPD, ALL, PID, Extended INFO)

netstat -tulpe
查看你的哪些进程正在监听着哪些端口

487 ☆arp

显示本地数据库
-H 指定硬件类别 ax25, ether,or pronet

可以查看重复的 ip
用-d 删除主机 panda 所有信息
arp -d panda

arp -s panda 00:00:c0:cf:a1:33
添加信息

488 ☆ping

ping 命令用于查看网络上的主机是否在工作，它向该主机发送 ICMP ECHO_REQUEST 包。有时我们想从网络上的某台主机上下载文件，可是又不知道那台主机是否开着，就需要使用 ping 命令查看。

语法：
ping [选项] 主机名/IP 地址

命令中各选项的含义如下：
-c 数目 在发送指定数目的包后停止。
-d 设定 SO_DEBUG 的选项。
-f 大量且快速地送网络封包给一台机器，看它的回应。

-I 秒数 设定间隔几秒送一个网络封包给一台机器，预设值是一秒送一次。
-l 次数 在指定次数内，以最快的方式送封包数据到指定机器（只有超级用户可以使用此选项）。
-q 不显示任何传送封包的信息，只显示最后的结果。
-r 不经由网关而直接送封包到一台机器，通常是查看本机的网络接口是否有问题。
-s 字节数 指定发送的数据字节数，预设值是 56，加上 8 字节的 ICMP 头，一共是 64ICMP 数据字节。

通过下面的方法使系统对 ping 没有反应
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

489☆traceroute

路由跟踪工具

490☆tcpdump

tcpdump 命令用于监视 TCP/IP 连接并直接读取数据链路层的数据包头。您可以指定哪些数据包被监视、哪些控制要显示格式。例如我们要监视所有 Ethernet 上来往的通信，执行下述命令：
tcpdump -i eth0

即使是在一个相对平静的网络上，也有很多的通信，所以我们可能只需要得到我们感兴趣的那些数据包的信息。在一般情况下，TCP/IP 栈只为本地主机接收入站的数据包绑定同时忽略网络上的其它计算机编址（除非您使用的是一台路由器）。当运行 tcpdump 命令时，它会将 TCP/IP 栈设置为 promiscuous 模式。该模式可接收所有的数据包并使其有效显示。如果我们关心的只是我们本地主机的通信情况，一种方法是使用“-p”参数禁止 promiscuous 模式，还有一种方法就是指定主机名：
tcpdump -i eth0 host 主机名

此时，系统将只对名为 hostname 的主机的通信数据包进行监视。主机名可以是本地主机，也可以是网络上的任何一台计算机。下面的命令可以读取主机 hostname 发送的所有数据：
tcpdump -i eth0 src host 主机名

下面的命令可以监视所有送到主机 hostname 的数据包：
tcpdump -i eth0 dst host 主机名

我们还可以监视通过指定网关的数据包：
tcpdump -i eth0 gateway Gatewayname

如果你还想监视编址到指定端口的 TCP 或 UDP 数据包，那么执行以下命令：
tcpdump -i eth0 host 主机名 and port 80

该命令将显示从每个数据包传出的头和来自主机 hostname 对端口 80 的编址。端口 80 是系统默认的 HTTP 服务端口号。如果我们只需要列出送到 80 端口的数据包，用 dst port;如果我们只希望看到返回 80 端口的数据包，用 src port。

491 ☆route

route add -net 127.0.0.0
这个命令将向路由表中添加一个指定地址或者网络的路由。注意此时网络为 A 类地址，掩码被设置为 255.0.0.0，这个新添加的条目被连接到 lo 设备上。

route add -net xxx.xxx.xxx.xxx netmask 255.255.255.0 dev eth0
这个命令为 IP 地址为 xxx.xxx.xxx.xxx 的主机增加一个路由，它的网络掩码被设置为 255.255.255.0。

route del -net xxx.xxx.xxx.xxx
此命令将删除 xxx.xxx.xxx.xxx 这个网络的路由。

route
网络的路由表
[root@lee /root]#route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.10.8.224 * 255.255.255.255 UH 0 0 0 eth0
10.10.8.0 * 255.255.255.0 U 0 0 0 eth0
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
default dgc8.njupt.edu 0.0.0.0 UG 0 0 0 eth0
default dgc8.njupt.edu 0.0.0.0 UG 1 0 0 eth0

- Destination 表示路由的目标 IP 地址。
- Gateway 表示网关使用的主机名或者是 IP 地址。上面输出的"*"表示没有网关。
- Genmask 表示路由的网络掩码。在把它与路由的目标地址进行比较之前，内核通过 Genmask 和数据包的 IP 地址进行按位"与"操作来设置路由。
- Flags 是表示路由的标志。可用的标志及其意义是：U 表示路由在启动，H 表示 target 是一台主机，G 表示使用网关，R 表示对动态路由进行复位设置;D 表示动态安装路由，M 表示修改路由，！表示拒绝路由。
- Metric 表示路由的单位开销量。
- Ref 表示依赖本路由现状的其它路由数目。
- Use 表示路由表条目被使用的数目。
- Iface 表示路由所发送的包的目的网络。

492 ☆DHCP

DHCP 服务的配置与应用
DHCP(Dynamic Host Configuration Protocol，动态主机配置协议)是一个简化主机 IP 地址分配管理的 TCP/IP 协议。

DHCP 服务的简介
DHCP 基于客户/服务器模式，当 DHCP 客户端启动时，它会自动与 DHCP 服务器通信，由 DHCP 服务器为 DHCP 客户端提供自动分配 IP 地址的服务。
安装了 DHCP 服务软件的服务器称为 DHCP 服务器，而启用了 DHCP 功能的客户机称为 DHCP 客户端。

DHCP 服务器是以地址租约的方式为 DHCP 客户端提供服务的
它有以下两种方式

1.限定租期
当 DHCP 客户端向 DHCP 服务器租用到 IP 地址后，DHCP 客户端可以使用该 IP 地址一段时间，当租约快到期时，客户端必须向 DHCP 服务器提出续约请求，请求成功后可以继续使用该 IP 地址。如果客户端没有续约或续约不成功，服务器就会将该 IP 地址回收，分配给其他 DHCP 客户端使用。当然原 DHCP 客户端之后还需要 IP 地址的话，它可以向 DHCP 服务器重新租用其他地址。这种方式是一种动态分配的方式，可以很好地解决 IP 地址不够用的问题。

2.永久租用
当 DHCP 客户端向：DHCP 服务器租用到 IP 地址后，这个地址就永久地分配给这个 DHCP 客户端使用(它是通过客户端网卡上的 MAC 地址来识别的)了。采用这种方式的前提是公司中的 IP 地址足够使用，这样 DHCP 客户端就不必频繁地向 DHCP 服务器提出续约请求。

493 ☆DHCP 服务工作原理

1.向 DHCP 服务器索取新的 IP 地址
当作为 DHCP 客户端的计算机第一次启动时，它需要经过一系列步骤才能获得其 TCP/IP 配置信息，并得到 IP 地址的租期

(1)DHCPDISCOVER(DHCP 发现)
该过程也被称为 IPDISCOVER。

当出现以下情况中的任意一种时，即启动 DHCP 发现过程。
当客户端第一次以 DHCP 客户端的身份启动，也就是它第一次向 DHCP 服务器请求 TCP/IP 配置时
该 DHCP 客户端所租用的 IP 地址已被 DHCP 服务器收回，并已提供给其他的 DHCP 客户端使用，而该 DHCP 客户端重新申请新的 IP 租约时
DHCP 客户端自己释放掉原先所租用的 IP 地址，并且要求租用一个新的 IP 地址时

客户端从固定 IP 地址方式转向使用：DHCP 方式时

在 DHCP 发现过程中，DHCP 客户端发出 TCP/IP 配置请求时，
DHCP 客户端使用 0.0.0.0 作为自己的 IP 地址，
255.255.255.255 作为服务器的 IP 地址，
然后以 UDP 的方式，在 67 或 68 端口广播出一个 DHCPDISCOVER 信息，
该信息含有 DHCP 客户端网卡的 MAC 地址和计算机的 NetBIOS 名称。
当第一个 DHCPDISCOVER 信息发送出去后，DHCP 客户端将等待 1 秒钟的时间。如果在此期间内没有 DHCP 服务器对此做出响应，DHCP 客户端将分别在第 9 秒、第 13 秒和第 16 秒时重复发送一次 DHCPDISCOVER 信息。如果仍然没有得到 DHCP 服务器的应答，DHCP 客户端就会在以后每隔 5 分钟广播一次 DHCP 发现信息，直到得到一个应答为止。

(2)DHCPOFFER(DHCP 提供)
当网络中的任何一个 DHCP 服务器在收到 DHCP 客户端的 DHCPDISCOVER 信息后，对自身进行检查，如果该 DHCP 服务器能够提供空闲的 IP 地址，就从该 DHCP 服务器的 IP 地址池中随机选取一个没有出租的 IP 地址，然后利用广播方式提供给 DHCP 客户端。在还没有将该 IP 地址正式租用给 DHCP 客户端之前，这个 IP 地址会暂时“隔离”起来，以免再分配给其他 DHCP 客户端。提供应答信息是 DHCP 服务器发给 DHCP 客户端的第一个响应，它包含了 IP 地址、子网掩码、租用期(以小时为单位)和提供响应的 DHCP 服务器的 IP 地址。
(3)DHCPREQUEST(DHCP 请求)
当 DHCP 客户端收到第一个由 DHCP 服务器提供的应答信息后，就以广播的方式发送一个 DHCP 请求信息给网络中所有的 DHCP 服务器。在 DHCP 请求信息中包含已选择的 DHCP 服务器返回的 IP 地址。
(4)DHCPACK(DHCP 确认)
一旦被选择的 DHCP 服务器接收到 DHCP 客户端的 DHCP 请求信息后，就将已保留的这个 IP 地址标识为已租用，然后也以广播方式发送一个 DHCPACK 信息给 DHCP 客户端。该 DHCP 客户端在接收 DHCP 确认信息后，就完成了获得 IP 地址的整个过程。

2.更新 IP 地址租约
当一台 DHCP 客户机租到一个 IP 地址后，它会有一个租期。当一个租期需要续租时，它的工作过程如下。

(1)当 DHCP 客户端的 IP 地址使用时间达到租期的一半时，它就会向 DHCP 服务器发送一个新的 DHCPREQUEST，若服务器在接收到该信息后并没有可拒绝该请求的理由时，便会送一个 DHCPACK 信息。当 DHCP 客户端收到该应答信息后，就重新开始一个租用周期。

(2)当进行 IP 地址的续租过程中出现以下两种特例中的任意一种时，需要另外处理。

DHCP 客户端重新启动时
不管 IP 地址的租期是否到期，每次启动 DHCP 客户端，都会利用广播的方式，给网络中所有的 DHCP 服务器发送一个 DHCPREQUEST 信息，以便继续使用原来的 IP 地址及其配置。如果此时

没有 DHCP 服务器对此做出应答，且原来 DHCP 客户端的租期还没有过期，那么 DHCP 客户端还将继续使用该 IP 地址。

IP 地址的租期超过一半但续约失败时
当续租失败后，DHCP 客户端仍然可以继续使用原来的 IP 地址及其配置，但是该 DHCP 客户端将在租期到达 87.5%的时候再次利用广播方式发送一个 DHCP 请求信息，以便找到一台可以继续提供租期的 DHCP 服务器。如果仍然续租失败，则该 DHCP 客户端会立即放弃其正在使用的 IP 地址，以便重新向 DHCP 服务器获得一个新的 IP 地址(重新申请就需要进行完整的 4 个过程)。在以上的续租过程中，如果续租成功，DHCP 服务器就会给该 DHCP 客户端发送一个 DHCPACK 信息，DHCP 客户端在收到该信息后进入一个新的 IP 地址租用周期;当续租失败时，DHCP 服务器将会给该 DHCP 客户端发送一个 DHCPNACK 信息，DHCP 客户端在收到该信息后，说明该 IP 地址已经无效或被其他 DHCP 客户端使用。

494 ☆DHCP 的安装

Red Hat Enterprise Linux 安装程序默认没有安装 DHCP 服务
rpm -q dhcp

495 ☆DHCP 启动停止

service dhcpd restart

496 ☆dhcpd.conf

DHCP 服务(即 dhcpd 守护进程)是按照/etc/dhcpd.conf 进行配置运行的。

默认情况下，该文件是不存在的，用户可以自行创建。

此外，其实在安装 DHCP 服务时都会安装一个范本文件，该文件的路径是
/usr/share/doc/dhcp-3.0.1/dhcpd.conf.sample
在具体的 DHCP 服务配置中，可将该文件复制为/etc/dhcpd.conf，然后根据需要进行编辑，这样设置比较容易。

在 dhcpd.conf 中
#为注释符号，其后该行的所有文字均为注释信息或无效的配置内容。
除括号那一行外，其他的每一行后面都要以“;”作为结尾

配置文件的格式
DHCP 配置文件 dhcpd.conf 的格式

选项/参数　　#这些选项/参数全局有效
声明{

选项/参数　　#这些选项/参数局部有效
}

1.声明
用于定义网络布局、提供给客户端的 IP 地址等
share-network 名称 {...} 定义超级作用域
subnet 网络号 netmask 子网掩码 {...}　　定义作用域(或 IP 子网)
range 起始 IP 地址 终止 IP 地址 定义作用域(或 IP 子网)范围
host 主机名 {...}定义保留地址
group {...}　定义一组参数

2.参数
参数是必选的或控制 DHCP 服务器行为的值。
ddns-update-style 类型　　定义所支持的 DNS 动态更新类型(必选)
allow/ignore client-updates 允许/忽略客户机更新 DNS 记录
default-lease-time 数字　　指定默认的租约期限
max-lease-time 数字 指定最大租约期限
hardware 硬件类型 MAC 地址　　指定网卡接口类型和 MAC 地址
server-name 主机名　通知 DHCP 客户机服务器的主机名
fixed-address IP 地址 分配给客户端一个固定的 IP 地址

注意：ddns-update-style、allow/ignore client-updates 这两个参数只能用于全局

3.选项
选项是用来配置 DHCP 客户端的可选参数，它们全部用 option 关键字作为开头。
subnet-mask 子网掩码　　为客户端指定子网掩码
domain-name “域名”　为客户端指定 DNS 域名
domain-name-servers IP 地址　为客户端指定 DNS 服务器的 IP 地址
host-name “主机名”　　为客户端指定主机名
routers IP 地址　为客户端指定默认网关
broadcast-address 广播地址　　为客户端指定广播地址
netbios-name-servers IP 地址　为客户端指定 WINS 服务器的 IP 地址
netbios-node-type 节点类型　　为客户端指定节点类型
ntp-server IP 地址　　为客户端指定网络时间服务器的 IP 地址
nis-servers IP 地址　　为客户端指定 NIS 域服务器的地址
nis-domain “名称”为客户端指定所属的 NIS 域的名称
time-offset 偏移差　　为客户端指定与格林尼治时间的偏移差

注意:以上的选项既可以用于全局,也可以用于局部

dhcpd.conf 文件如下
ddns-update-style interim;

```
ignore client-updates;

subnet 192.168.152.0 netmask 255.255.255.0
{
range 192.168.152.129 192.168.152.254;

option routers 192.168.152.128;
option subnet-mask 255.255.255.0;

option nis-domain "panda.com";
option domain-name "panda.com";
option domain-name-servers 192.168.152.128,192.168.152.1

option time-offset                -18000; # Eastern Standard Time
#option ntp-servers                 192.168.1.1;
#option netbios-name-servers       192.168.1.1;

option broadcast-address 192.168.16.255;

#range dynamic-bootp 192.168.0.128 192.168.0.254;  #只用于本地网络可删去 dynamic-bootp
default-lease-time 86400;   #1 天
max-lease-time 172800;     #2 天

host pcl{
      hardware ethernet 00:aO:CC:cf:9C:14;
      fixed-address 192.168.16.20;
}

host pc2{
      hardware ethernet 04:2O:C1:f0:9C:11;
      fixed-address 192.168.16.30;
}
}
```

497 ☆设置 IP 作用域

IP 作用域是一个 IP 子网中所有可分配的 IP 地址的连续范围，必须在 DHCP 服务器内设置一个 IP 作用域。当 DHCP 客户端向 DHCP 服务器请求 IP 地址时，DHCP 服务器就可以从该作用域内选择一个尚未分配的 IP 地址，并将其分配给该 DHCP 客户端。

在 dhcpd.conf 文件中,用 subnet 来声明 IP 作用域
subnet 子网 ID netmask 子网掩码

{
range 起始 IP 地址 结束 IP 地址;　　#指定可分配给用户的 IP 地址范围
IP 参数;　　#定义客户端的 IP 参数,如子网掩码,默认网关等
}

值得注意的是，语句中的子网 ID 必须与 DHCP 服务器所在的网络 ID 相同。

通常一个 IP 作用域对应一个 IP 子网段。IP 子网段可用一个或多个 range 语句来描述，但是多个 range 所定义的 IP 范围不能重复
range 192.168.152.128 192.168.16.100;
range 192.168.16.101 192.168.16.200;

498 ☆设置客户端的 IP 选项

除了给 DHCP 客户端指定 IP 地址外，还可以利用 DHCP 服务器来设置客户端的工作环境，例如，可以设置其子网掩码、DNS 服务器的地址和默认网关等。当 DHCP 客户端向 DHCP 服务器索取 IP 地址或更新租约时，DHCP 服务器就会自动为 DHCP 客户端设置这些 IP 选项。

在配置文件 dhcpd.conf 中，设置 DHCP 客户端 IP 选项的语句的基本格式为：
option 选项代码 设置内容

1.设置路由器(默认网关)的 IP 地址
可用 option routers 语句来设置 DHCP 客户端的默认网关的地址
option routers 192.168.152.128;

2.设置子网掩码
可用 option subnet-mask 语句来设置 DHCP 客户端的子网掩码。
option subnet-mask 255.255.255.0

3.设置 DNS 域名
可用 option domain-name 来设置 DHCP 客户端的 DNS 域名
option domain-name "panda.com"

4.设置 DNS 服务器的 IP 地址
用 option domain-name-server 语句来设置 DHCP 客户端的 DNS 服务器的 IP 地址。
option domain-name-servers 192.168.152.128,192.168.152.1
可设定主 DNS 和第二 DNS

5.广播地址
可用 option broadcast-address 语句来设置 DHCP 客户端在该 IP 子网中的广播地址
option broadcast-address 192.168.16.255;

499 ☆设置租约期限

租约期限是在 DHCP 服务器上指定的时间长度，在这个时间范围内 DHCP 客户端可以临时使用从 DHCP 服务器租借到的 IP 地址。如果客户端在租约到期前，没有更新租约，则 DHCP 服务器会收回该 IP 地址，并将该 IP 地址提供给其他的 DHCP 客户端使用。如果原 DHCP 客户端之后还需要 IP 地址的话，它就可以向 DHCP 服务器重新申请租用另一个 IP 地址。
当然，DHCP 客户端可以在租约未到期前更新租约。实际上，每当 DHCP 客户端重新启动时，以及经过租约期限的 1/2 后，都会主动向 DHCP 服务器要求更新租约。只要能够成功更新租约，DHCP 客户端就可以继续使用其 IP 地址，并且会重新取得一个新的租约。

在 dhcpd.conf 文件中，有下面两个与租约期限有关的设置。
1.默认的租约期限
可用 default-lease-time 语句来设置默认的租约时间长度，其单位为秒
default-lease-time 86400; #1 天

2.最大租约期限
可用 max-lease-time 语句来设置客户机租用 IP 地址的最长时间
max-lease-time 172800;　　#2 天

一旦网络上的 DHCP 客户端开始从 DHCP 服务器中租用了 IP 地址，租用信息就会被记录在 /var/lib/dhcp/dhcpd.leases 文件中，该文件会不断被更新.从该文件中就可查看到 IP 地址分配情况，包括每个租用的 IP 地址及对应的 MAC 地址、租约的起始时间和结束时间等信息。

tail -f /var/lib/dhcp/dhcpd.leases
查看 dhcp 更新记录

500 ☆保留特定的 IP 地址

可以保留特定的 IP 地址给指定的 DHCP 客户端使用，也就是说，当这个客户端每次向 DHCP 服务器索取 1P 地址或更新租约时，DHCP 服务器都会给该客户端分配相同的 IP 地址。这种 DHCP 服务器为 DHCP 客户端分配 IP 地址的方式，通常也被称为静态分配 IP 或固定分配 IP。

要保留特定的 IP 地址给指定的 DHCP 客户端使用，可先用 arp 命令查出该客户端网卡的 MAC 地址;然后在/etc/dhcpd.conf 文件中，加入如下格式的 host 语句
host 主机名 {
hardware ethernet　网卡的 MAC 地址;　　#指定 DHCP 客户端网卡的 MAC 地址
fixed-address　IP 地址;　　#指定为该 DHCP 客户端分配的 IP 地址
IP 参数;　　#指定默认网关等其他 IP 参数
}

先 ping -c 3 192.168.152.131
看是否在线

用 arp 看刚才 ping 的 ip 所对应的 MAC 地址(获得 MAC 地址)
最后在/etc/dhcpd.conf 中加
host pcl{
    hardware ethernet 00:0C:29:B0:B2:EB;
    fixed-address 192.168.152.131;
}

501 ☆分配多网段的 IP 地址

如果在一个规模比较大的物理网络中存在多个 IP 子网，而多个 IP 子网的主机都需要 DHCP 服务器来提供地址配置信息，那么可以采用的一种方法是：在每一个 IP 子网中都安装一台 DHCP 服务器，让它们分别为各个子网分配 IP 地址。但是，从节约成本的角度出发，一般情况下都不这样做，而是采用在一个子网中安装 DHCP 服务器并让它来为多个子网分配 IP 地址的方法。那么如何才能使该 DHCP 服务器向多个网段提供动态 IP 分配服务呢?这时可以考虑使用 DHCP 中继代理功能。DHCP 中继代理(dhcrelay)允许将无 DHCP 服务器的子网内的 DHCP 客户请求转发给其他子网内的一个或多个 DHCP 服务器。

1.在 DHCP 服务器上设置超级作用域
修改 DHCP 服务器上的 dhcpd.conf 文件
shared-network 名称 {
    subnet 子网 1 的网络 ID netmask 子网掩码{
    }
    subnet 子网 2 的网络 ID netmask 子网掩码{
    }
}
这里的 shared-network 语句用于向 DHCP 服务器声明某些 IP 子网(如子网 1、子网 2)、实际上是共享一个物理网络

ddns-update-style interim;
ignore client-updates;
shared-network mynet{
option subnet-mask 255.255.255.0;
option domain-name "panda.com";
option domain-name-servers 192.168.16.2，61.144.56.100
option broadcast-address 192.168.16.255;
default-lease-time 86400;
max-lease-time 172800;

subnet 192.168，16.0 netmask 255.255.255.0
{
range 192.168.152.128 192.168.16.100;
option routers 192.168.16.1;

```
host pcl{
      hardware ethernet 00：aO：CC：cf：9C：14;
      fixed-address 192.168.16.20;
}
host pc2{
      hardware ethernet 04：2O：C1：f0：9C：11;
      fixed-address 192.168.16.30;
}
}

subnet 192.168，17.0 netmask 255.255.255.0
{
range 192.168.17.10 192.168.17.100;
option routers 192.168.17.1;
}

}
```

值得注意的是，由于部分选项(如子网掩码、DNS 服务器的 IP 地址等)的设置对于 2 个子网来说都是相同的，因此，将它们放在 subnet 语句的括号外边，这样这部分选项对 3 个子网都会起作用。但是，由于默认网关这个选项对于 2 个子网来说是不相同的，因此在 2 个子网各自的 subnet 语句括号内分别设置该选项。

此外，由于 ddns-update-style interim(指定 DHCP 服务器支持的 DNS 动态更新的方式为 interim)和 ignore client-updates(指定不允许客户机更新 DNS 记录)是对整个 DHCP 服务器及所有子网都有效的全局参数，因此应将它们设置在最外层。

502 ☆DHCP 中继代理

在连接 3 个子网的那台计算机上安装 DHCP 中继代理很容易，只要安装 DHCP 服务就行了，因为安装 DHCP 服务时就自动安装上了 DHCP 中继代理 dhcrelay。

一般情况下，DHCP 中继代理监听所有接口上的 DHCP 请求，当然也可以向某个子网指定 DHCP 中继代理。实例中，由于 DHCP 服务器位于网络接口为 eth0 的子网 A 内，就可以用 DHCP 中继代理只向 eth1 和 eth2 连接的子网 B、C 内提供 DHCP 服务，具体实现方法是编辑 /etc/sysconfig/dhcrelay 配置文件，将语句 INTERFACES=“”改为 INTERFACES=“ethl eth2”，同时将语句 DHCPSERVERS=“”改为 DHCPSERVERS=“192.168.16.177”具体配置内容如下

```
INTERFACES="ethl eth2"#只向 eth1 和 eth2 连接的子网提供 DHCP 服务
DHCPSERVERS="192.168.16.177"      #子网 A(eth0 上)的 DHCP 服务器
```

此外，也可以通过执行下面的命令来实现

dhcrelay -i eht1 -i eht2 192.168.16.177

除了使用 DHCP 中继代理外，还可以通过选用具有 DHCP/BOOTP 中继功能(定义在 RFC 1542)的路由器，以支持 DHCP 服务器实现跨网段分配 IP 地址。

503 ☆DHCP 客户端的配置

Linux 中 DHCP 客户端的配置

dhclient
直接从 dhcp 中获取地址
/etc/dhclient.config
默认没有,同样拷贝过来
# dhclient.conf
interface "eth0"{
prepend domain-name-servers 127.0.0.1;
request subnet-mask, broadcast-address, time-offset, routers,
domain-name, domain-name-servers, host-name;
require subnet-mask, domain-name-servers;
}
# end dhclient.conf

或者
①直接编辑文件/etc/sysconfig/network-scripts/ifcfg-ethO，找到语句"BOOTPROTO=none"将其改为“BOOTPROTO=dhcp”即可
②重新启动网卡，可在该客户端的本地终端窗口下执行下面的命令：
ifdown eth0;ifup eth0
或
ifconfig eth0 down;ifconfig eth0 up
③测试该 DHCP 客户端是否已配置好，可执行下面的命令
ifconfig eth0

Windows 2000/XP/2003 中 DHCP 客户端的配置
选自动获得 IP 和自动获得 DNS
测试 ipconfig /all
更新租约 ipconfig /renew
释放 IP ipconfig /release

504 ☆DNS

网络中为了区别各个主机，必须为每台主机分配一个惟一的地址，这个地址即称为“IP 地址”。但这些数字难于记忆，所以就采用“域名”的方式来取代这些数字。不过最终还是必须将域名转

换为对应的 IP 地址才能访问主机。通过建立 DNS 数据库，记录主机名称与 IP 地址的对应关系，驻留在服务器端，为客户端的主机提供 IP 地址解析服务。当某台主机要与其他主机通信时，就可以利用主机名称向 DNS 服务器查询该主机的 IP 地址。整个 DNS 域名系统由以下 4 个部分组成。

1.DNS 域名空间
指定用于组织名称的域的层次结构，它如同一棵倒立的树，层次结构非常清晰，根域位于顶部，紧接着在根域的下面是几个顶级域，每个顶级域又可以进一步划分为不同的二级域，二级域再划分出子域，子域下面可以是主机也可以是再划分的子域，直到最后的主机。在 Internet 中的域是由 InterNIC 负责管理的，域名的服务则由 DNS 来实现。
2.资源记录
将 DNS 域名映射到特定类型的资源信息，以便在名称注册和解析时使用
3.DNS 服务器
提供存储和应答资源记录的名称查询服务。
4.DNS 客户端
用来查询 DNS 服务器，将域名称解析为查询中指定的资源记录类型。

505 ☆DNS 查询的工作原理

当 DNS 客户端需要查询所使用的名称时，它会查询 DNS 服务器来解析该名称。客户端发送的查询消息包括以下 3 条信息。
指定的 DNS 域名，必须为完全合格的域名(FQDN)
指定的查询类型，可根据类型指定资源记录，或者指定为查询操作的专门类型
DNS 域名的指定类别

DNS 查询以各种不同的方式进行解析。客户端可以使用从先前的查询获得的缓存信息就地应答查询；
DNS 服务器也可使用其自身的资源记录信息缓存来应答查询；
DNS 服务器还可以代表请求客户的查询或联系其他 DNS 服务器，以便完全解析该名称，并将应答返回至客户端，这个过程称为递归。
客户端本身也可尝试联系其他的 DNS 服务器来解析名称。然后根据来自 DNS 服务器的参考答案，使用其他的独立查询，该过程称作迭代。

总之，DNS 查询过程按两部分进行：
• 名称查询从客户端计算机开始，并传送给本机的 DNS 客户服务程序进行解析；
• 如果不能在本机解析查询，可根据设定的查询 DNS 服务器来解析名称。

以上两种查询方式的具体工作过程如下。

本地解析
当客户端提出解析请求时，首先将请求传送至 DNS 客户服务,以便使用本地缓存信息进行解析，如果可以解析所要查询的名称，则 DNS 客户服务应答该查询，该请求处理过程结束。

本地解析程序的缓存包括两种名称信息：
本地配置的主机文件，该文件是主机名称到地址的映射信息，在 DNS 客户服务启动时预先加载到缓存中；
从以前的 DNS 查询应答的响应中获取的资源记录，它被保留在缓存中一段时间。

如果此查询与本机的缓存中的项目不匹配，则解析过程将继续进行，客户端将查询 DNS 服务器来解析名称。

查询 DNS 服务器
当客户端请求无法在本地解析时，将请求发送至 DNS 服务器。DNS 服务器接收到查询请求时，首先检查它能否在服务器的本地配置区域中获取资源记录信息做出应答

如果查询的名称与本地区域信息中的相应资源记录匹配，则使用该信息来解析查询的名称，服务器做出应答，此次查询完成。
如果区域信息中没有查询的名称，则服务器检查它能否通过来自先前查询的本地缓存信息来解析该名称。如果从中发现了匹配的信息，则服务器使用该信息应答查询，此次查询完成。
如果无论从缓存还是从区域信息，查询的名称在首选服务器中都未发现匹配的应答，那么查询过程可继续进行，使用递归来完全解析名称。

如要递归查询 center.panda.com 的地址，首选 DNS 服务器通过分析完全合格的域名，向顶层域 com 查询，而 com 的 DNS 服务器与 panda.com 服务器联系以获得更进一步的地址，这样循环查询直到获得所需要的结果，并一级级向上返回查询结果，最终完成查询工作。

需要注意的是，为了让 DNS 服务可以正确运行递归查询，需要有一些必要的信息，该信息通常以根目录的形式来提供，借助使用根目录提示寻找根域服务器，DNS 服务器可以完成递归查询。

如果客户端申请使用递归过程，但在 DNS 服务器上禁用递归或查询 DNS 服务器时客户端没有申请使用递归，则使用迭代的方式查询

如要迭代查询 user.center.panda.com 的地址，首先 DNS 服务器在本地查询不到客户端请求的信息时，就会以 DNS 客户端的身份向其他配置的 DNS 服务器继续进行查询，以便解析该名称。在大多数情况下，可能会将搜索一直扩展到 Intenet 上的根域服务器，但根域
服务器并不会对该请求进行完整的应答，它只会返回 example.corn 服务器的 IP 地址，这时 DNS 服务就根据该信息向 panda.com 服务器查询，　由 panda.com 服务器完成对 user.center.panda.com 域名的解析后，再将结果返回 DNS 服务器。

对于大多数的迭代查询而言，如果它的主 DNS 不能解析该名称，那么客户端会使用本地配置的 DNS 服务器列表,在整个 DNS 名称空间中联系其他名称服务器

DNS 规划

在选择域名时必须符合 RCF 1123 中的规定：所有大写字母(A~Z)、小写字母(a~z)、数字(0~9)和连字符(-)

506 ☆Bind 的简介

Linux 下架设 DNS 务器通常是使用 Bind 程序来实现，Bind 是 Berkeley Internet Name Domain Services 的简写
Bind 原本是美国 DARPA 资助研究伯克里大学(Berkeley)开设的一个研究生课题，后来经过多年的变化发展，已经成为世界上使用最为广泛的 DNS 服务器软件，目前 Internet 上绝大多数的 DNS 服务器都是用 Bind 来架设的。

Bind 能够运行在当前大多数的操作系统平台之上。目前 Bind 软件由因特网软再
联合会 ISC(Intemet Software Consonium)这个非赢利性机构负责开发和维护。
ISC 的官方网站(http://www.isc.org/)

507 ☆DNS 服务的安装

Red Hat Enterprise Linux 安装程序默认没有安装 DNS 服务
需要 bind,bind-utils 和 caching-nameserver 软件包
rpm -q bind
rpm -q bind-utils
rpm -q caching-nameserver

安装 chroot 软件包
chroot 是 Change Root 的缩写，它可以改变程序运行时所参考的“/”根目录位置，即将某个特定的子目录作为程序的虚拟“/”根目录。chroot 对程序运行时可以使用的系统资源、用户权限和所在目录进行严格控制，程序只在这个虚拟的根目录具有权限，一旦跳出该目录就无任何权限了，所以有些书籍也将 chroot 称为“jail 监禁”

rpm -q bind-chroot

使用了 chroot 后，由于 Bind 程序的虚拟根目录是/var/nanmed/chroot,所以下文提到所有的 DNS 服务器配置文件、区域数据文件和配置文件内的语句，都是相对这个虚拟根目录而言，如下文提到的/etc/named.conf，其真正的路径是是/var/nanmed/chroot/etc/named.conf；如目录/var/named/,真正的路径是/var/named/chroot/var/named

508 ☆named

/etc/sysconfig/named
当 dns 服务的以 chroot 模式运行，你可以修改下/etc/sysconfig/named 下的
ROOTDIR=/var/named/chroot 该成 ROOTDIR=/ 就行了。那样就能以正常模式的 dns 工作方式工作了。

509 ☆DNS 防火墙设置

DNS 查询通常使用的是 UDP 端口 53，如果当传输的数据大于 512kB 和服务器之间区域复制操作时，则使用 TCP 端口 53。如果 Linux 服务器开启了防火墙功能。就需关闭防火墙功能或设置允许 UDP 和 TCP 端口 53 通过。可以使用以下命令开放 UDP 和 TCP 端口 53：
iptables -I INPUT -p udp --dport 53 -j ACCEPT
iptables -I INPUT -p tcp --dport 53 -j ACCEPT

510 ☆主配置文件

Bind 的主配置文件是/etc/named.conf,该文件只包括 Bind 的基本配置，并不包含任何 DNS 区域数据。安装 DNS 服务后，安装程序会自动生成 named.conf 文件的默认内容。

named.conf 文件格式有如下规则：
配置文件中语句必须以分号结尾；
需用花括号将容器指令(如 options)中的配置语句包含起来；
注释符号可以使用 C 语言中的符号对“/*"和 ¨ */”、c++语言的“//”和 Shell 脚本的"#"

设置服务器的选项
容器指令 options 内的语句都属于定义服务器的全局选项，这个语句在每个配置文件中只有一处。如果出现多个 options 语句，则第一个 options 的配置有效，并且会产生一个警告信息。如果没有 options 语句，每个选项使用缺省值。

(1)设置 named 要读写文件的路径
directory 选项定义了 DNS 区域数据文件的路径，配置文件中后续的语句如果只指定文件名，而没给出文件的路径(如 file "named.ca";)，则文件的路径默认为此处定义的路径。
directory "/var/named"

(2)设置服务器存放数据库的路径
dump-file 选项定义服务器存放数据库的路径，当执行 rndc dumpdb 命令时会用到该选项。
dump-file "/var/named/data/cache_dump.db";

(3)设置服务器统计信息文件的路径
statistics-file 选项定义了服务器统计信息文件的路径，当使用 rndc stats 命令时会用到该选项。
statistics-file ”/var/named/data/named_stats.txt”；

rndc 命令的作用
rndc 命令是 Bind 9 远程名称守护进程配置工具(在 Bind 8 中称为 ndc)，它可以用来启动、停止、重载配置文件，转储其状态，转入调试模式和获得服务器状态信息，等等。rndc 的“r”是 remote 的意思，这意味者 mdc 还可以控管到远程的 DNS 服务器。
controls {

inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
限定 rndc 命令限定为本地

(4)设置共享密匙文件
include 选项可以将一个外部配置文件包含到主配置文件中，包含后的外部配置文件语句与主配置文件语句的效果是一样的。rndc.key 共享密匙文件用来简化从 Bind 8 系统上的升级过程，这样 Bind 9 就可以继续使用和 Bind 8 一样的配置文件。
include ”/etc/rndc.key";

rndc-confgen -a -b 512
生成一个/etc/rndc.key 新密钥

(5)定义服务器的版本信息
version 选项定义用户向服务器请求版本信息时，服务器返回的版本信息。默认返回的是服务器的真实版本。　　。
version "4.9.10"

黑客在攻击前，一般都会查询目标 DNS 服务器的版本信息，以便利用该版本的漏洞进行攻击。为此，在上例中定义虚假的“4.9.10”版本来误导黑客。

511 ☆主要域名服务器

主要名称服务器用于存放该区域中相关设置的 DNS 服务器。主要名称服务器存放的是区域文件的正本数据,而且以后当这个区域的数据要更改时，也是直接修改主要名称服务器存放的区域文件。

512 ☆设置根区域

当 DNS 服务器处理递归查询时，如果本地区域文件不能进行查询的解析，就会转到根 DNS 服务器查询，所以在主配置文件 named.conf 文件中还要定义根区域。
"."区域是 DNS 层次中的最高层。根服务器提供了哪些服务器对于给定的域享有授权。"."节应该以如下的方式出现:
zone "." IN {
type hint;
file "named.ca";
};

这里对各行语句分别进行说明。
(1)设置根区域
容器指令 zone 的作用是定义一个 DNS 区域，指令后面跟着 DNS 区域的名称，根区域的名称,根区域的名称是"."

在容器指令 zone 花括号内定义该 DNS 区域的选项
zone "." IN {
};

(2)设置区域的类型为“hint”
type 选项定义了 DNS 区域的类型，根区域“.”应该设置为“hint"，这样当服务器启动的时候，它能找到根 DNS 服务器并得到根 DNS 服务器的最新列表
type hint;

(3)设置根服务器列表文件名
file 选项定义保存区域数据的文件名。虽然文件名用户可以自行定义，但为了管理方便，该文件一般命名为“named.ca”。每一个标准的 DNS 服务器都有一个保存根服务器列表的文件，它包括 internet 上的根服务器对应的 IP 地址。Bind 根据该文件得到根 DNS 服务器的 IP 地址

513 ☆设置正向解析区域

/etc/named.conf 中保存正向查询区域.主区域 DNS 服务器保存着某个区域(如 panda.com)的数据信息。
zone "panda.com" IN {
type master;
file "panda.com.zone";
allow-transfer {
192.168.152.129;
192.168.7.177;
};
};

(1)设置主区域的名称
容器指令 zone 后跟着的是主区域的名称,表示这台 DNS 服务器保存着 panda.com 区域的数据,网络上其他所有 DNS 客户机或 DNS 服务器都可以通过这台 DNS 服务器查询到与这个域相关的信息。
zone "panda.com" IN {
};

(2)设置类型为主区域
type 选项定义了 DNS 区域的类型，对于主区域应该设置为"master"类型。
type master;

(3)设置主区域文件的名称
file 选项定义了主区域文件的名称.一个区域内的所有数据(如主机名和对应的 IP 地址、刷新间隔,过期时间)必须放在区域文件中。虽然文件名用户可以自行定义。为了方便管理,文件名一般是区域的名称,扩展名是.zone
file "panda.com.zone";

(4)设置辅域名服务器的地址
allow-transfer 选项定义了容许进行区域复制的辅域名服务器地址
为了安全起见，必须严格限制区域复制操作.即指定只能向信任服务器的辅域名服务器开放区域复制功能。如果没有辅域名服务器，可将语句该为"allow-transfer {none; }; "，从而禁止区域复制功能。

```
allow-transfer {
192.169.152.129;
192.168.7.17;
};
```

514 ☆设置反向解析区域

在大部分的 DNS 查询中，DNS 客户端一般执行正向查找，即根据计算机的 DNS 域名查询对应的 IP 地址。但在某些特殊的应用场合中(如判断 IP 地址所对应的域名是否合法)，也会使用到通过 IP 地址查询对应 DNS 域名的情况(也称为反向查找)。下面是一个在/etc/named.conf 文件定义反向解析区域的例子。

```
zone "152.168.192.in-addr.arpa" IN {
type master;
file "152.168.192.zone";
allow-transfer{
192.169.152.129;
192.168.7.17;
};
};
```

(1)设置反向解析区域的名称
容器指令 zone 后面跟着是反向解析区域的名称。在 DNS 标准中定义了固定格式的反向解析区域 in-addr.arpa，以便提供对反向查找的支持。

```
zone "16.168.192.in-addr.arpa" IN {
};
```

与 DNS 名称不同，当从左向右读取 IP 地址时，它们是以相反的方式解释的，所以需要将域中的每个 8 位字节数值反序排列。从左向右读 IP 地址时，读取顺序是从地址的第一部分最一般的信息(IP 网络地址)到最后 8 位字节中包含的更具体的信息(IP 主机地址)。
因此，创建 in-addr.arpa 域中的子域时候，应该按照带点的十进制表示法编号的 IP 地址的相反顺序组成的。如子网“192.168.16.0/24”完整的反向解析域名为“16.168.192.in.addr.arpa”，子网“192.168.0.0/16”完整的反向解析域名为“168.192.in-addr.arpa”。

(2)设置区域的类型为“master”
type 选项定义了 DNS 区域的类型，由于反向解析区域属于一种比较特殊的主区域，所以还是设置为“master”类型。

type master;

(3)设置反向解析区域文件的名称
file 选项定义了反向解析区域文件的名称。虽然文件名用户可以自行定义，但为了方便管理，文件名一般是需反向解析的子网名，扩展名是.arpa。
file "192.168.16.arpa"

(4)设置辅域名服务器的地址
反向解析区域同样也可以进行区域复制
allow-transfer{
192.169.152.129;
192.168.7.17;
};

515 ☆辅助域名服务器

辅助名称服务器也可以向客户机提供域名解析功能，但它与主要名称服务器不同的是，它的数据不是直接输入的，而是从其他服务器(主要名称服务器或其他的辅助名称服务器)中复制过来的，只是一份副本，所以辅助名称服务器中的数据无法被修改。
当启动辅助名称服务器时，它会和与它建立联系的所有主要名称服务器建立联系，并从中复制数据。在辅助名称服务器工作时，还会定期地更改原有的数据，以尽可能地保证副本与正本数据的一致性。在一个区域中设置多台辅助名称服务器具有以下优点。
-提供容错能力。当主要名称服务器发生故障时，由辅助名称服务器提供服务。
-分担主要名称服务器的负担。在 DNS 客户端较多的情况下，通过架设辅助名称服务器完成对客户端的查询服务，可以有效地减轻主要名称服务器的负担。
-加快查询的速度,实现异地子站点的分布

一个从域名服务器将为一个区域提供授权的回答,但不是区域的授权开始。

现在将重新配置您的域名服务器作为 panda.com 区域和 152.168.192.in-addr.arpa 区域的从域名服务器。

在/etc/named.conf 文件中添加如下行
zone "panda.com" IN {
type slave;
masters { 192.168.152.128; };
file "slaves/panda.com.zone";
};

zone "152.168.192.in-addr.arpa" IN {
type slave;
masters { 192.168.152.128; };

file "slaves/192.168.152.zone";
};

(1)设置从区域
容器指令 zone 后面跟着的是从区域的名称，从区域的名称要与主要名称服务器上的主区域名称相同。
zone "panda.com" IN {
};

(2)设置类型为从区域
type 选项定义了 DNS 区域的类型，对于从区域应该设置为“slave”类型。
type slave;

(3)设置主区域文件的名称
file 选项定义了从区域文件的名称。从区域文件的数据不需要管理员直接输入，而是从其他服务器中复制生成，只是一份副本。如果不需要辅助名称服务器保存区域数据的备份，则可以删除该行。
file "slaves/panda.com.zone";

注意:Bind 已经建立了一个专门用于存放从区域文件的目录/var/named/chroot/var/named/slaves/，所以在上例设置主区域文件的路径是“slaves/panda.com.zone”，切勿自行指定路径，否则会造成 DNS 服务启动出错，查看/var/log/messages 曰志时可以看到会出现“permission denied”的错误

(4)设置主要名称服务器的地址
masters 选项定义了主区域服务器的域名或 IP 地址，辅助名称服务器启动时或达到刷新时间间隔时会自动连接主区域服务器并复制其中的 DNS 数据。
masters { 192.168.152.128; };

只有在主要名称服务器允许当前可以进行区域传输的情况下，辅助名称服务器才能进行区域复制操作。

516 ☆配置仅有缓存的服务器

Cache-only 服务器是很特殊的 DNS 服务器，它本身并不管理任何区域，但是 DNS 客户端仍然可以向它请求查询。Cache-only 服务器类似于代理服务器，它没有自己的域名数据库，而是将所有查询转发到其他 DNS 服务器处理，当 Cache-only 服务器收到查询结果后，除了返回给客户机外，还会将结果保存在缓存中。当下一个 DNS 客户端再查询相同的域名数据时，就可以从高速缓存里查出答案，加快 DNS 客户端的查询速度。如果在局域网中建立一台这样的：DNS 服务器，就可以提高客户机 DNS 的查询效率并减少内部网络与外部网络的流量。

这种类型的域名服务器对于任何区域都不授权。
仅有缓存的域名服务器被设定为主域名服务器。当主机名称或者 IP 地址需要被解析的时候，仅有缓存的域名服务器将查询请求转发到另外一台域名服务器或者到根域名服务器来决定授权的用来解析的域名服务器。
一旦解析完成，仅有缓存的域名服务器在缓存中存储解析的信息，该解析信息有一段的生存周期。以后的查询将会变得很快。

架设 Cache-only 服务器非常简单，只需要建立好主配置文件 named.conf 即可，架设 Cache-only 服务器的主配置文件/etc/named.conf 也需要设置服务器的选项，方法与设置主要名称服务器的方法相同，这里就不再重复了。下面通过一个配置 Cache-only 服务器/etc/named.conf 文件的完整例子来说明各项配置。

在 options 中添加:
```
forward only;
forwarders {
    192.168.152.128;
    192.168.152.129;
};
```

这将导致您工作站上的仅有缓存的域名服务器转发其不能解析的 DNS 查询到在 192.168.152.128 的域名服务器，并且如果超时，不与根域名服务器直接联系。
仅仅是本机 DNS 中不存在的域名才转发!!!!!!!!

(1)设置转发方式
forward 选项定义了请求转发的方式，通常将其设置为 only，表示服务器只把客户机的查询转发到其他 DNS 服务器上去。此选项只有当 forwarders 列表中有内容的时候才有意义。
```
forward only;
```
也可以为 forward first;

(2)设置转发到哪些 DNS 服务器
容器指令 forwarders 定义了如果此 DNS 不能解析出 IP 就将客户机的查询转发到哪些 DNS 服务器，可以添加多个 DNS 服务器的地址，Cache-only 服务器首先将查询转发给使用第 1 台 DNS 服务器，如果第 1 台 DNS 服务器没有应答，则会将查询转发给第 2 台 DNS 服务器，依此类推，直到接收到来自 DNS 服务器的确定应答。
```
forwarders {
    192.168.152.128;
    192.168.152.129;
};
```
这样的格式也可以的{ 210.35.92/24; 192.168.0.254 ;}

也可以定义到一个区域中:
```
zone "panda.com" IN {
```

type forward; //类型是转发
forwarders {
    192.168.152.128;
    192.168.152.129;
};

517 ☆DNS 区域数据库文件

您的主要配置文件指定了/var/named 为区域数据库所在的目录。(如果有 chroot 则为 /var/named/chroot/var/named)

格式:
[domain] [class] <type> <rdata>

[domain]
如: panda.panda.com. ,www
www 是简写,实际会默认添上定义这个文件的那个 zone ,即:www=www.panda.com.这个段可以省略不写,默认跟前一行的[domain]相同.
@表示"panda.com."

[class]
一般为 internet 类,即:IN ,可以省去不写.

<type>
分别有 SOA,NS,A,MX,CNAME,及反向数据库里的 PTR.
SOA=Start Of Authority. 必要的
NS=nameserver. 必要的
A=域名和 ip 地址的映射. 必要的
MX=mail exchange. 可选的
CNAME=域名的一个别名. 可选的
PTR=用在反向 ip 地址和域名的映射. 必要的

假名不可以放在 NS MX 的<rdata>字段.

SOA 括弧中的 5 个数字:
对应为(serial refresh retry expire Minimum)
serial 版本,可以是任何数字,一般:年月日修改数.
refresh 从服务器更新时间,是 SOA 信息的刷新时间.
Retry 从服务器重新更新时间,是与授权服务器联系的频率.
Expire slave 从服务器数据保留时间,服务器保存有关的区域信息,不更新它的时间间隔.
Minimum 最小 TTL 值,区域中记录的存活时间.

H=hour D=day M=minute W=week

SOA 后的 panda.panda.com.表示这个区域授权给哪台机.
root.panda.panda.com.表示管理人.用.代替@
每一个记录文件只能有一个 SOA ,不能重复，而且必须是所负责的 zone 中第一个"记录".

接着 SOA 后面，指定了这个区域的授权主机和管理者的信箱，这里分别是@和“root”，也就是 localhost.主机和 root 信箱。这里要注意的是：SOA 的主机名称必须能够在 DNS 系统中找到一个 A 记录 (以后会提到);另外，我们平时使用的信箱通常是“user@host”这样的格式，但因为@在 dns 记录中是个保留字符(刚才已经提过)，所以在 SOA 中就用“.”来代替了@。目前这个信箱是 root(并没有主机地址)，也就是本机，您可以写成“root.localhost.”但不能写成“root@localhost.”。

接下来的 SOA 设置，是被括在“( )”之间的 5 组数字，主要作为和 slave 服务器同步 dns 资料所使用的资料：
Serial：其格式通常会是“年月日+修改次序”(但也不一定如此，您自己能够记得就行)。当 slave 要进行资料同步的时候，会比较这个号码。如果发现在这里的号码比它那边的数值“大”，就进行更新，否则忽略。不过设 serial 有一个地方您要留意：不能超过 10 位数字！
Refresh：这里是是告诉 slave 要隔多久要进行资料同步(是否同步要看 Serial 的比较结果)。
Retry：如果 slave 在进行更新失败后，要隔多久再进行重试。
Expire：这是记录逾期时间：当 slave 一直未能成功与 master 取得联系，那到这里就放弃 retry，同时这里的资料也将标识为过期(expired)。
Minimum：这是最小默认 TTL 值，如果您在前面没有用“$TTL”来定义，就会以此值为准。

以上的数字都是以秒为单位，但您也可以用 H(小时)、D(天)、W(星期)来做单位，如：3H 和 259200 是一样的。

请注意：SOA 记录中这对“()”符号之第一个“(”括号一定要和 SOA 写在同一行，而不能用 Enter 断行到下一行去，而且其左边最好有一个空格键或 tab 建。而最后一个“)”括号也不能写在注解符号“;”的右边。

518 ☆根服务器信息文件 named.ca

/var/named/named.ca 是一个非常重要的文件，该文件包含了 Internet 的根服务器名字和地址，Bind 接收到客户端主机的查询请求时，如果在 Cache 中找不到相应的数据，就会通过根服务器进行逐级查询。

如，当服务器收到来自 DNS 客户机查询 www.panda.com 域名的请求时，如果 Cachc 中没有相应的数据，就会向于 Internet 的根服务器请求，然后根服务器将查询交给负责域.com 的(SOA)权威名称服务器，域.com 的(SOA)权威名称服务器再将请求交给负责域 panda.com 的(SOA)权威名称服务器进行查询。

由于 named.ca 文件经常会随着根服务器的变化而发生变化，所以建议最好从国际互联网络信息中心(InterNIC)的 FTP 服务器下载最新的版本，下载地址为 ftp://ftp.rs.internic.net/domain/named.root。下载完后，应将该文件改名为 named.ca，并复制到“/var/named/chroot/var/named/”目录下。

dig -t NS .
可以产生 named.ca 文件(先设好 resolv.conf 的 nameserver)

519 ☆正向解析区域文件

一个区域内的所有数据(包括主机名和对应 IP 地址、刷新间隔和过期时间等)必须存放在 DNS 服务器内，而用来存放这些数据的文件就被称为区域文件(区域数据文件使用“;”符号注释)。DNS 服务器的区域数据文件一般存放在/var/named/目录下，一台 DNS 服务器内可以存放多个区域文件，同一个区域文件也可以存放在多台 DNS 服务器中。下面是一个在 /var/named/chroot/var/named/panda.com.zone 文件的完整例子。

主机名称可以是一个完全符合标准的名称(FQDN)，也可以是一个缩写。
所有的不以点号结尾的主机名称都将被视为缩写，并且区域名称被附加到主机名称的后面。

```
$TTL 86400
@               IN SOA    panda.panda.com. root.panda.panda.com. (
                                        42              ; serial (d. adams)
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

@               IN NS             panda

@ IN A 192.168.152.128
panda IN A 192.168.152.128
askpanda IN A 192.168.152.129
www IN A 192.168.152.128
ftp IN A 192.168.152.128
smtp IN A 192.168.152.128
pop IN A 192.168.152.128
pop3 IN A 192.168.152.128
imap IN A 192.168.152.128
imaps IN A 192.168.152.128

mail1.panda.com.    IN    A    192.168.152.128
mail2.panda.com.    IN    A    192.168.152.129
mail3.panda.com.    IN    A    192.168.152.130
```

www1 IN CNAME panda
www2 IN CNAME panda
www3 IN CNAME panda

panda.com. IN MX 10 mail1.panda.com.
panda.com. IN MX 11 mail2.panda.com.
panda.com. IN MX 12 mail3.panda.com.

(1)设置允许客户端缓存来自查询的数据的默认时间
$TTL 选项定义了允许客户端缓存来自查询的数据的默认时间，单位是秒，通常应将它放在文件的第 1 行。如果数据不是经常变动的，可以考虑把它的值设为几天，否则设置得太小(如几个小时)会引起不必要的 DNS 查询流量。该数值对所有在该域中的记录有效
$TTL 86400

区域文件的时间数字默认都是以秒为单位，为了方便理解，也可以用 h(小时)、d(天)和 w(星期)来做单位，36000 和 10h 的表示方式是一样的。

(2)设置起始授权机构 SOA 资源记录
SOA 是 Start of Authority(起始授权机构)的缩写，它是主要名称服务器区域文件中必须要设定的资源记录，它表示创建它的 DNS 服务器是主要名称服务器。SOA 资源记录定义了域名数据的基本信息和其他属性(更新或过期间隔)。通常应将 SOA 资源记录放在区域文件的第 1 行或紧跟在$TTL 选项之后。

```
panda.com.          IN SOA    panda root.panda.panda.com. (
                                    2006081303          ; serial (d. adams)
                                    3H                  ; refresh
                                    15M                 ; retry
                                    1W                  ; expiry
                                    1D )                ; minimum
```

设置所管辖的域名
“panda.com."定义了当前 SOA 所管辖的域名(注意域名后有一个句点号“.”)，当然也可以不使用域名而使用符号“@”来代替。不过为了管理方便，建议使用域名。

另外:
$ORIGIN panda.com.
$ORIGIN（一定要大写）设定,说明下面的记录出自何处.请您加倍留意最后的一个小小数点"."
特殊字符@,建议不要和 $ORIGIN 同时使用。它就是 $ORIGIN 的意思,也就是刚刚所定义的 $ORIGIN panda.com 内容
沒有定义 $ORIGIN 的话，那這个@的值就是 named.conf 里的 zone .

设置 Internet 类
"IN"代表类型是属于 Internet 类，这个格式是固定不可改变的。

设置授权主机名
例子中，“panda.panda.com.”定义了负责该区域的名称解析的授权主机名，这样 DNS 服务器才会知道谁控制这个区域。
授权主机名称必须在区域文件中有一个 A 资源记录。(panda IN A 192.168.152.128)

“panda”主机号(末尾没有句点号)资源记录使用了相对名称，Bind 会自动在其后面的“.panda.com”，此规则对区域文件中的所有语句都适用。

注意在完整的主机地址末尾加上一个句点号，以表示这是一个完整的主机名，这是因为任何末尾没加句点号“.”的名称都会被视为本区域内的相对域名，如“www.panda.com”(末尾没有句点号)会当成“www.panda.com.panda.com”解析。此规则对区域文件中的所有语句都适用。

设置负责该区域的管理员的 E-mail 地址
例子中，"root.panda.panda.com."定义了负责该区域的管理员的 E-mail 地址，由于在 DNS 中使用符号"@"代表本区域的名称，所以在 E-mail 地址应使用句点号“.”代替“@”。

设置 SOA 资源记录的各种选项
小括号"()"里的数字是 SOA 资源记录各种选项的值，主要作为和辅助名称服务器同步 DNS 数据而设置的。
注意“(”号一定要和 SOA 写在同一行。

设置序列号
“2006081303”定义了序列号的值，它的格式通常是“年月日+修改次数”(当然也从 0 开始，然后在每次修改完主区域文件后使这个数加 1，而且不能超过 10 位数字。序列号用于标识该区域的数据是否有更新，当辅助名称服务器需要与主要名称服务器进行区域复制操作(即同步辅助名称服务器的 DNS 数据)的时候，就会比较这个数值。如果发现在这里的数值比它最后一次更新时的数值大，就进行区域复制操作，否则放弃区域复制操作，所以每次修改完主区域文件后都应增加序列号的值。

设置更新间隔
例子中，“3H”定义了更新间隔的值。更新间隔用于定义辅助名称服务器隔多久时间与主要名称服务器进行一次区域复制操作。

设置重试间隔
例子中，“15M”定义了重试间隔的值。重试间隔用于定义辅助名称服务器在更新间隔到期后，仍然无法与主要名称服务器取得联系时，重试区域复制的间隔。通常该间隔应小于更新间隔。

设置过期时间

例子中，“1W”定义了过期时间的值。过期时间用于定义辅助名称服务器在该时间内一直不能与主要名称服务器取得联系时，则放弃重试并丢弃这个区域的数据(因为这些数据有可能会失效或错误)

设置最小默认 TTL
例子中，“1D”定义了最小默认 TTL 的值。最小默认 TTL 定义允许辅助名称服务器缓存查询数据的默认时间，如果文件开头没有“$TTL”选项则以此值为准。

(3)设置名称服务器 NS 资源记录
名称服务器 NS(Name Server)资源记录定义了该域名由哪个 DNS 服务器负责解析，NS 资源记录定义的服务器称为区域权威名称服务器。权威名称服务器负责维护和管理所管辖区域中的数据，它被其他服务器或客户端当作权威的来源，为 DNS 客户端提供数据查询，并且能肯定应答区域内所含名称的查询
@ IN NS panda

正如您对于"panda.com"只能有一个单一的域名服务器，您也只能有一个单一的 NS 记录。

(4)设置主机地址 A 资源记录
主机地址 A(Address)资源记录是最常用的记录，它定义了 DNS 域名对应 IP 地址的信息。地址(A)记录将主机名称映射到 IP 地址(域名服务器的主要功能)。
在上面的例子中，使用了两种方式来定义 A 资源记录，一种是使用相对名称，另外一种是使用完全规范域名 FQDN(Fully Qualified Domain Name)，这两种方式只是书写形式不同而已，在使用上没有任何区别：对于相对名称 panda，Bind 会自动在相对名称的后面加上后缀“.panda.com”，所以相当于完全规范域名的 panda.panda.com.。
@ IN A 192.168.152.128
panda IN A 192.168.152.128
askpanda IN A 192.168.152.129
www IN A 192.168.152.128
ftp IN A 192.168.152.128
smtp IN A 192.168.152.128
pop IN A 192.168.152.128
pop3 IN A 192.168.152.128
imap IN A 192.168.152.128
imaps IN A 192.168.152.128

注意第一个 A 记录设定了域的"缺省的 IP 地址"。
接下来的 A 记录建立了多个主机名称对应一个 IP 地址。

(5)设置别名 CNAME 资源记录
别名 CNAME(Canonical Name)资源记录也被称为规范名字资源记录。CNAME 资源记录允许将多个名称映射到同一台计算机上，使得某些任务更容易执行

访问 www2.panda.com 和 www3.panda.com 时，实际都是访问 www1.panda.com，即 IP 地址为 192.168.152.128 的计算机。
www1 IN CNAME panda
www2 IN CNAME panda
www3 IN CNAME panda

别名(CNAME)记录建立了主机名称的别名。注意到别名映射到主机名称而不是 IP 地址。
CNAME 不应该出现在右边的数据区域作为真实的主机名称，对于多重别名的解析的速度会很慢。

(6)设置邮件交换器 MX 资源记录
邮件交换器 MX(Mail eXchanger)资源记录指向一个邮件服务器，用于电子邮件系统发邮件时根据收信人邮件地址后缀来定位邮件服务器。例如当一个邮件要发送到地址 panda@panda.com，邮件服务器通过 DNS 服务器查询 panda.com 这个域名的 MX 资源记录，如果 MX 资源记录存在，邮件就会发送到 MX 资源记录所指定的邮件服务器上。
可以设置多个 MX 资源记录，指明多个邮件服务器，优先级别由 MX 后的数字决定，数字越小，邮件服务器的优先权越高。优先级高的邮件服务器是邮件传送的主要对象，当邮件传送给优先级高的邮件服务器失败时，可以把它传送给优先级低的邮件服务器。

```
panda.com.      IN   MX   10 mail1.panda.com.
panda.com.      IN   MX   11 mail2.panda.com.
panda.com.      IN   MX   12 mail3.panda.com.
```

由于 MX 资源记录只登记了邮件服务器的域名，而在邮件实际传输时，是通过邮件服务器的 IP 地址进行通信的，所以邮件服务器还必须在区域文件中有一个 A 资源记录，以指明邮件服务器的 IP 地址，否则会导致传输邮件失败的错误。

邮件交换记录(MX)记录了一个主机它将会处理给定的域或者主机邮件的转发。当一个邮件传递代理(MTA)试图投递信件的时候，它将首先试图在 DNS 中查找目的主机的 MX 记录。如果该 MX 记录存在，那么将直接发送到 MX 记录指定的主机。反之，如果不存在 MX 记录，MTA 对于目的主机进行标准的 DNS 查询，并且直接投递到该主机上去。MX 记录用来建立邮件的网关，和作为缺省的对于域的邮件的目的地。

520☆反向解析区域文件

反向解析区域文件的结构和格式与区域文件类似，只不过它的主要内容是建立 IP 地址映射到 DNS 域名的指针 PTR 资源记录。
下面是一个在/Var/named/chroot/var/named/152.168.192.zone 文件定义反向解析区域的例子。

```
$ttl 36000
152.168.192.in-addr.arpr. IN SOA panda.panda.com. root.panda.com. (
                                        2006081303          ; serial (d. adams)
                                        3H                  ; refresh
```

15M ; retry
1W ; expiry
1D) ; minimum

IN NS panda.panda.com.

128.152.168.192.in-addr.arpr. IN PTR panda.panda.com.

128 IN PTR www.panda.com.
128 IN PTR mail1.panda.com.
129 IN PTR mail2.panda.com.
130 IN PTR mail3.panda.com.

(1)设置 SOA 和 NS 资源记录
反向解析区域文件同样必须包括的 SOA 和 NS 资源记录，使用固定格式的反向解析区域 in-addr.arpa 作为域名，结构和格式与区域文件类似，这里就不再重复了。

注意在 NS 记录开头的空白的地方是非常特别的，并且被解释为"和上一条记录相同"的缩写。在本例中，上一条记录为符号"@"，其本身就是在主配置文件中定义的域名的缩写。

区域文件中,所有开头的空白,都是表示和上一条记录相同

(2)设置指针 PTR 资源记录
指针 PTR 资源记录只能在反向解析区域文件中出现，PTR 资源记录和 A 资源记录正好相反，它是将 IP 地址解析成 DNS 域名的资源记录。与区域文件的其他资源记录类似，它也可以使用相对名称和完全规范域名 FQDN，如对于资源记录“128 IN PTR www.panda.com.”，Bind 会自动在其后面的“.152.168.192.in-addr.arpa”，所以相当于完全规范域名的 128.152.168.192.in-addr.arpa。

128.152.168.192.in-addr.arpa. IN PTR panda.panda.com.

9 IN PTR www.panda.com.
178 IN PTR mail1.panda.com.
179 IN PTR mail2.panda.com.
180 IN PTR mail3.panda.com.

指针 PTR 资源记录，注意最后一条资源记录只有主机号(末尾没有句点号)，Bind 会自动在其后面的“.152.168.192.in-addr.arpa.”

指针(PTR)记录通过间接的机制将名称映射到 IP 地址。作为分离的技术来进行 IP 地址的反查询的替代，BIND 采用了一种修改的对于特定主机名称的正向查询的方式。这种"反向域名查询"以反转的 IP 地址后面添加"in-addr.arpa"域的形式出现。这将允许域名服务器使用相同的机制进行正反两方面的查询。

521 ☆实现负载均衡功能

DNS 负载均衡的优点是经济简单易行，它在 DNS 服务器中为同一个域名配置多个 IP 地址(即为一个主机名设置多条的 A 资源记录)，在应答 DNS 查询时，DNS 服务器对每个查询将以 DNS 文件中主机记录的 IP 地址按顺序返回不同的解析结果，将客户端的访问引导到不同的计算机上去，使得不同的客户端访问不同的服务器,从而达到负载均衡的目的.

在企业网中需要使用 3 台内容相同的 FTP 服务器共同承担客户对网站的访问,它们的 IP 地址分别对应 192.168.152.128,192.168.152.129 和 192.168.152.130.现只要在 DNS 服务器的区域文件中加入以下 3 条 A 资源记录,就可以实现 3 台 FTP 服务器网络负载均衡功能.

```
ftp   IN   A    192.168.152.128
ftp   IN   A    192.168.152.129
ftp   IN   A    192.168.152.130
```

为了解析 FTP 请求响应所有这些资源记录，DNS 服务器会轮询这些资源记录，以随机的顺序来响应用户的解析请求，实现了将客户机的访问分担到每个 FTP 服务器上的负载均衡功能，
提示: 试图尝试设定这些的 A 记录的 TTL 为 0

522 ☆实现直接解析域名

许多用户有直接使用域名访问 Web 网站的习惯,即在浏览器中不输入主机名 WWW,而直接使用如 http://baidu.com 或 http://tom.com/来访问.然而,并不是所有的 Web 网站都支持这种访问方式,只有 DNS 服务器能直接解析域名的网站才可以

DNS 服务器默认只能解析完全规范域名 FQDN,不能直接将域名解析成 IP 地址造成的.为了方便用户访问,可以在 DNS 服务器的区域文件加入下面一条特殊的 A 资源记录,以便支持实现直接解析域名功能.

```
panda.com.      IN      A       192.168.152.128
```

或

```
        .       IN      A       192.168.152.128
```

完成后,用户只要访问域名 panda.com,DNS 服务器就会将其直接解析成 IP 地 192.168.152.128。理解了句点号的作用，就能明白以上两种形式的作用是一样的。

523 ☆实现泛域名的解析

泛域名是指一个域名下的所有主机和子域名都被解析到同一个 IP 地址上
如使用命令“ping panda.panda.com”会发现 DNS 服务器能解析 panda.panda.com 这个域名，然后再使用“ping wawa.panda.com”也可以发现 DNS 服务器能解析这个域名

不难发现，在域名 panda.com 前加上任意主机名，DNS 服务器都可以解析到同一个 IP 地址上，这是因为负责解析 panda.com 的 DNS 服务器使用了泛域名技术。泛域名在实际使用中的作用是非

常广泛的，除了可以将泛域名解析到默认的 Web 网站以方便用户的访问外，还可以实现基于数据幸的二级域名管理。

可以在 DNS 服务器的区域文件末尾(注意是末尾)加入下面一条特殊的 A 资源记录(符号“*”是代表任何
字符的通配符)，以便支持实现泛域名解析功能。
*.panda.com.　　IN　　A　　192.168.152.128
或
*　　IN　　A　　192.168.152.128

524 ☆基于数据库的二级域名管理

很多企业都为员工架设了个人 Web 站点来满足工作的需要，为了节省费用，这些 Web 网站通常采用虚拟主机技术，即在同一个服务器上架设多个网站。员工使用二级域名访问这些站点。由于员工数比较多，所以在 DNS 服务器上维护这些二级域名的工作量非常大。采用泛域名解析技术就可以很好地解决这个难题。即将泛域名解
析到一台 Web 服务器上，Web 服务器上的默认主页程序(如 index.cgi、index.php 和 index.jsp)根据对用户浏览器发送的 HTTP 访问请求头信息进行分析，并分割出二级子域名，然后查询数据库得到子域名对应的 URL，并利用重定向技术将访问的用户带到目标地，如网易免费二级域名转向系统 yeah.net 就是这样实现的。

525 ☆DNS 客户端的配置

在 Linux 中配置 DNS 客户端的方法很简单，可直接编辑文件/etc/resolv.conf，然后使用 nameserver 选项来指定 DNS 服务器的 IP 地址，

可以使用 nameserver 选项来指定多达 3 台 DNS 服务器。如果指定了 3 台以上的 DNS 服务器，则只有前 3 台 DNS 服务器有效。客户端按照 DNS 服务器在文件中的顺序进行查询的，如果没有接收到 DNS 服务器的响应，就去尝试向下一台服务器查询，直到试完所有的服务器为止，所以应该将速度最快、最可靠的 DNS 服务器列在最前面，以保证在查询时不会超时。

526 ☆DNS 重启

重启 dns 服务不是 service named restart : 正确是： service named reload (或/etc/init.d/named reload)
或者
rndc reload

527 ☆DNS 测试

named-checkconf (必须主机登陆，telnet,su 等登陆都不执行)
named-bootconf
named-checkzone

pidof named

host <你要解析的 IP 或域名>
或
nslookup <你要解析的 IP 或域名>

测试域名服务器
host panda
host panda.panda.com
dig panda.panda.com
dig panda.panda.com@192.168.152.128
dig panda.com
host 192.168.152.128
dig -x 192.168.152.128
dig -x 192.168.152.128
host www
host www1

记住 dig 期望给与一个 FQDN 作为查询，
然而 host 则通过查看位于文件/etc/resolv.conf 的查询信息。试着在别的人的域名服务器和子域上进行附加的查询。如果设定正确，您将能够在其他教室系统上进行正向和反向查询。

查看一下服务器添加到/var/log/messages 文件中的条目。确定您的域名在调入的时候没有发生错误。

528☆nslookup

语法：
nslookup [IP 地址/域名]

nslookup 程序是 DNS 服务的主要诊断工具，它提供了执行 DNS 服务器查询测试并获取详细信息。使用 nslookup 可以诊断和解决名称解析问题、检查资源记录是否在区域中正确添加或更新，以及排除其他服务器相关问题。nslookup 命令的功能是查询一台机器的 IP 地址和其对应的域名。它通常需要一台域名服务器来提供域名服务。如果用户已经设置好域名服务器，就可以用这个命令查看不同主机的 IP 地址对应的域名。

nslookup 有两种运行模式：非交互式和交互式。

非交互式通常用于返回单块数据的情况，其命令格式：
nslookup [-选项] 查询的域名 [DNS 服务器地址]

如果没有指明 nslookup 要使用 DNS 服务器地址，则 nslookup 使用/etc/resolv.conf。文件定义的 DNS 服务进行查询。非交互式 nslookup 程序运行完后，就会返回 Shell 提示符下。如果要查询另外一条记录，则需要重新执行该程序，

交互式通常用于返回多块数据的情况，其命令格式：
nslookup [- DNS 服务器地址]

如果没有指明 nslookup 要使用 DNS 服务器地址，则 nsookup 使用/etc/resolv.conf。文件定义的 DNS 服务进行查询。运行交互式 nslookup 程序，就会进入 nslookup 程序提示符“>”，接下来就可以在“>”后输入 nslookup 的各种命令、需查询的域名或反向解析的 IP 地址。查询完一条记录可接着在“>”后输入新的查询，使用 exit 命令可退出 nslookup 程序.由于对 DNS 服务器进行测试往往需要连续查询多条记录，所以实际更多地使用 nslookup 程黲的交互式

测试主机地址 A 资源记录
进入 nslookup 程序后，默认的查询类型是主机地址，在 nslookup 程序提示符“>”下直接输入要测试的完全规范域名 FQDN，nslookup 会显示当前 DNS 服务器的名称和 IP 地址，然后返回完全规范域名 FQDN 对应的 IP 地址

测试反向解析指针 PTR 资源记录
在 nslookup 程序提示符“>”下直接输入要测试的 IP 地址，nslookup 会返回 IP 地址所对应的完全规范域名 FQDN

测试别名 CNAME 资源记录
在 nslookup 程序提示符“>”下先使用命令“set type=cname”设置查询的类型为别名，然后输入要测试的别名，nslookup 会返回对应的真实计算机

测试邮件交换器 MX 资源记录
在 nslookup 程序提示符“>”下先使用命令“set type=mx”设置查询的类型为邮件交换器然后输入要测试的域名，nslookup 会返回对应的邮件交换器地址

测试起始授权机构 SOA 资源记录
在 nslookup 程序提示符“>”下先使用命令“set type=soa”设置查询的类型为起始授权机构然后输入要测试的域名，nslookup 会返回对应的 SOA 资源记录内容

测试名称服务器 NS 资源记录
在 nslookup 程序提示符“>”下先使用命令“set type=ns”设置查询的类型为名称服务器，然后输入要测试的域名，nslookup 会返回对应的名称服务器地址

测试负载均衡
测试负载均衡需要在查询的类型为主机地址，如果当前的查询类型不是主机地址，就应在 nslookup 程序提示符“>”下先使用命令“set type=a”设置查询的类型为主机地址，然后输入要测试的负载均衡完全规范域名 FQDN，nslookup 会返回对应的所有 IP 地址

测试直接解析域名
测试直接解析域名需要在查询的类型为主机地址，如果当前查询类型不是主机地址，应在 nslookup 程序提示符“>”下先使用命令“set type=a”设置查询的类型为主机地址，然后输入要测试的直接解析域名，nslookup 会返回域名对应的 IP 地址

测试泛域名
测试泛域名需要在查询的类型为主机地址，如果当前查询类型不是主机地址，就应在 nslookup 程序提示符“>”下先使用命令“set type=a”设置查询的类型为主机地址，然后输入任意主机名的域名(没有相应的 A 记录)，对于每个任意主机名的域名，nslookup 会返回同一个 IP 地址

测试外部 Internet 域名
除了要测试本地 DNS 服务器的区域数据外，还要测试 DNS 服务器是否能解析外部：Internet 的域名(需要保证 DNS 服务器能与 Internet 连接)，即测试 named.conf 文件定义的根区域是否正确。在。nslookup 程序提示符“>”下输入如 www.baidu.com 等 Internet 上的完全规范域名 FQDN，nslookup 会返回对应的 IP 地址

529 ☆DNS 安全访问控制

named.conf 中
logging { //把错误日志定义到默认的/var/log/syslog 文件中,同时建议查看 messages 文件中的错误信息
category default { default_syslog; default_debug; };
};

//---首先定义各安全群组---//
acl "local" { //定义本机 IP 域
127.0.0.0/8;
};
acl "CNC_net" { //定义了网通的 dns 主机 IP 域，这样普通主机只能通过网通的转发来查询以增加安全性
202.0.0.0/8;
};
acl "TRUSTED_net" { //受信任的 IP 域
219.156.81.0/24;
192.168.2.0/24;
};
acl "TRUSTED_host" { //受信任的主机地址
192.168.100.26/32;
10.0.1.130/32;
10.0.1.161/32;
};

```
acl "VPN_net" { //定义了一个 VPN 网络 IP 区域
10.87.200.0/24;
};
acl "BAD_Guys" { //定义黑名单区域
221.15.164.250;
}; //-注：以上 ip 仅作例子

options {
directory "/var/named";
/*
 * If there is a firewall between you and nameservers you want
 * to talk to, you might need to uncomment the query-source
 * directive below.  Previous versions of BIND always asked
 * questions using port 53, but BIND 8.1 uses an unprivileged
 * port by default.
 */
// query-source address * port 53;
forward first;
forwarders {
 202.102.224.68;
 202.102.227.68;
 };

 allow-transfer { none; }; //默认禁止区域转移
 allow-query { any; }; //默认允许查询
 blackhole { BAD_Guys; }; //默认黑名单

};

//
// a caching only nameserver config
//
zone "." IN {
type hint;
file "caching-example/named.ca";
};

zone "localhost" IN {
type master;
file "caching-example/localhost.zone";
allow-update { none; };
};
```

```
zone "0.0.127.in-addr.arpa" IN {
type master;
file "caching-example/named.local";
allow-update { none; };
};
zone "hnzzcc.pn" IN {
type master;
file "db.hnzzcc.pn";
allow-query { VPN_net; }; //允许 VPN 网络查询
  allow-transfer { VPN_net; }; //允许 VPN 网络区域转移

};
zone "200.87.10.in-addr.arpa" IN {
            type master;
            file "db.10.87.200";
           allow-query { VPN_net; };//允许 VPN 网络查询
            allow-transfer { VPN_net; }; //允许 VPN 网络区域转移
};
zone "hnzzcc.com" IN {
            type master;
            file "db.hnzzcc.com";
            allow-query { local; CNC_net; TRUSTED_net;VPN_net; }; //允许本机，网通 dns，受信任网
络,VPN 查询
            allow-transfer { TRUSTED_host; }; //允许受信任网络区域转移
};
zone "217.137.211.in-addr.arpa" IN {
            type master;
            file "db.211.137.217";
            allow-query { local; CNC_net; TRUSTED_net;VPN_net; }; //允许本机，网通 dns，受信任网
络,VPN 查询
            allow-transfer { TRUSTED_host; }; //允许受信任网络区域转移
};

zone "ln.hnzzcc" IN { //定义转发区域 ln.hnzzcc
type forward;
forwarders { 10.87.13.87; 10.87.13.96; }; //转发地址
};
```

在设置 dns 的安全原则的时候，有些问题您必须注意：

如果 client (包括 localhost) 如果不在 allow-query 范围内的话，将不能查询该区域的任何信息。

当这一个区域作为 master 且有其它 slave 指向它的时候，slave 主机必须同时被包括在 allow-query 和 allow-transfer 设居中才可以完成区域转移。

那些不在 allow-query 设置当中的主机，虽然不能够直接将 server 指向这台 dns 来查询所在区域。然而，如果对方先将 server 指向另一台 dns 主机，且该主机是属于 allow-query 设置之中的话，也可以对该区域进行查询。

例如，如果在 zone "hnzzcc.com" 当中允许 CNC_net 的查询。虽然从外面的主机不能直接查询这个 zone ，但只要对方将 server 指向 CNC_net 其中任一台 dns 主机，而该主机的 allow-query 可以让其通过的话，那他们也就可以查询 hnzzcc.com 了。

530 ☆dnssec-keygen

[root@panda html]# dnssec-keygen -a HMAC-MD5 -b 128 -n user dnsadmin
之后会生成二个密钥文件 Knamed.+157+39224.key,Knamed.+157+39224.private

[root@panda html]# cat Knamed.+157+39224.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: Yu4GuBxkYA/0KjGZLeNc2g== #这个就是密钥了!

[root@root ~]#vi /var/named/chroot/var/named/kdnsadmin.key
Key dnsadmin{
Alogorithm HMAC-MD5.SIG-ALG.REG.INT;
Secrct "Yu4GuBxkYA/0KjGZLeNc2g==";
};

Include”/var/named/chroot/var/named/kdnsadmin.key” #定义密钥的位置!
在 allow-update {key dnsadmin;};写入

531 ☆DNS 的 DHCP 设置

至于完成 DNS 服务器的配置，还是不够的还要配置 DHCP 来协同动态 DNS 来工作!
[root@panda ~]# vi /etc/dhcp.conf
#
#dhcpd.conf
ddns-update-style interim; #开启动态 dns 功能!
Option domain-name "panda.com.";
Option domain-name-servers 192.168.152.128;
Default-lease-time 600;

Max-lease-time 800;

Subnet 10.1.1.0 netmask 255.255.255.0 {
range 10.1.1.10 10.1.1.254;
Option broadcast-address 10.1.1.255;
Option routers 10.1.1.2
}
Key dnsadmin
Algorithm HMAC-MD5.SIG-ALG.REG.INI;
Secret “4Gef1Mkmn5hrlwyueGJV3g==”;
}

Zone panda.com. {
Primary 192.168.152.128;
Key dnsadmin;
}

Zone 1.1.10.in-addr.arpa. {
Primary 10.1.2.1;
Key dnsadmin;
}

启动 DHCP 服务器
启动 DNS 服务器

532 ☆telnet

Telnet 服务有一个致命的弱点--它以明文的方式传输用户名及口令，所以，很容易被别有用心的人窃取口令。

533 ☆telnet 安装

需要
telnet-server-0.17-30
telnet-0.17-30

534 ☆telnet 启动停止

/etc/xinetd.d/telnet 文件里面的 disable=yes 改成 disable=no
然后 service xinetd restart

535 ☆telnet 防火墙

iptables -I INPUT -p tcp --dport 23 -j ACCEPT
netstat 只能看到 xinetd,不能 grep telnet

536 ☆telnet 配置

/etc/xinetd.d/telnet 中
instances=3
同时只允许 3 个连接

537 ☆telnet 端口

/etc/services 中
telnet 23/tcp
telnet 23/udp
改动后用新的端口

连接时用
telnet ip 或域名 端口号

538 ☆使用 telnet

telnet 服务器的 ip 地址或域名

539 ☆telnet 用 root 登录

telnet 默认不充许 root 用户登陆的

vi /etc/pam.d/login
在 auth required pam_securetty.so 前边加#,注释掉
这个方法在 EL4 下不可用,明文传 root 密码是很危险的

540 ☆SSH

Secure Shell，是一个难以置信地强大的工具。它允许你从一台 Unix 主机完全控另外一台。你能远程的执行程序，用一个屏的显示传递到本地的主机，始终安全的回避掉窃听者(获取有效信息)。安全壳有把你的键盘以及监视器封装起来(加密)建立连接的效果。

RSA,DSA 都是非对称加密方法，DES,3DES,blowfish 等都是对称加密方法，md5,sha 是 hash 单向算法;凡是用公钥加密，只能被相应私钥解开

SSH 客户端与服务器端通讯时，用户名及口令均进行了加密，有效防止了对口令的窃听。

网上要安全通信，关键是交换数据要加密，为何不用直接公钥加密，私钥解密呢？比如双方相互交换公钥，发送方用对方公钥加密要发送数据，接收方用相应私钥解密，很简单么?但是用公钥加密数据的计算量巨大，不适合批量数据，所以采取对话建立时利用公私钥来传送一个对称加密方法的 key，以后的通信都是用这个对称加密法和这个确定的 key 加密来进行的

具体大致如下(忽略协商版本等细节):设 Server Slient 要通信
1、C ----------> S
C 提出要和 S 加密通信

host_key.pub
cookie
2、C <---------- S
S 把自己的公钥(host_key.pub)、一个随机的 cookie 数给 C

加密的 session_key
3、C ----------> S
C 利用 cookie 和收到的公钥计算一个 session_id ,
C 自己产生一个 session_key,用公钥加密,传给 S

利用 session_key 加密的数据
4、C <----------> S
S 用私钥解开 session_key
S 用 cookie 和公钥同样计算 session_id，应该和 C 计算出的一致
S 用 session_key 和 C 开始在唯一 session 上加密数据通信

541 ☆SSH 版本

关于安全壳的日志中有多种安全壳版本 Versions(版本): ssh1(安全壳版本 1),ssh2(安全壳版本 2),以及 openssh 开放安全壳。
openssh 是从商业版本的安全壳中分离出来的安全壳版本--安全壳版本 2 是首选!
SSH 还分为 SSH1 及 SSH2 两个版本，SSH1 是一个完全免费的软件包，而 SSH2 在商业使用时则要付费。
SSH1 是第一版，它的功能没有 SSH2 强大，但是，由于它是免费的，所以广泛地使用在很多网站中。
SSH2 中加入了很多功能，并且兼容 SSH1 服务器，可以对 SSH1 的客户端提供很好的服务支持。

首先，SSH 软件包由两部分组成，一部分是服务器端软件包，另一部分是客户软件包。

windows 上的服务器软件包只能运行在 Windows NT 及 Windows 2000 Server 以上的版本中，而客户端 SSH 可以运行在所有的 Windows 系统中。

542 ☆SSH 安装

需要三个包
openssh-3.9p1-8.RHEL4.1
openssh-server-3.9p1-8.RHEL4.1
openssh-clients-3.9p1-8.RHEL4.1

openssh-server 和 openssh-clients 都依赖于 openssh

543 ☆SSH 生成密钥

先把/etc/ssh/sshd_config 中改成
PasswordAuthentication 要设置为 no,用密钥认证

生成一对密钥,公钥保存在 ssh 服务器用户主目录的.ssh 目录下的 authorized_keys 文件里,私钥保存在本地计算机中

产生密钥用
ssh-keygen -t rsa
ssh-keygen -t dsa
sh-keygen -b 2048 -t rsa (生成基于 RSA 加密算法的长度为 2048 位的密匙。)
要求用户输入一个长的认证字串，这个字串的功能同 password 相当，但是，它更长，一般是在 20 个字符以内。用于判断用户是否有使用密钥的权力。如果有,ssh 登录的时候会要求输入
Enter passphrase for key '/root/.ssh/id_rsa':输入密码
再次输入相同的字串以确认输入的正确，之后，系统产生一对密钥及公钥。将公钥复制到本地，以便客户端对服务器发送的信息进行解密用。当然，如果你不复制，在第一次登录时，服务器会将它的公钥自动推给客户机，以便客户机能对服务器提供的信息进行解密识别。

保存在当前用户的.ssh 目录下
私钥为 id_rsa
公钥为 id_rsa.pub

传递方法:
mail -s "my key" panda < ~/.ssh/id_dsa.pub

mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/bob": 1 message 1 new
>N 1 askpanda@askpanda Fri Sep 19 15:56 13/982 "my key"
& w panda_key
"alice.key" [New file]

& q

把公钥传输到服务器的用户主目录下.ssh 中,改名为 authorized_keys
cd ~/.ssh
cp id_rsa.pub authorized_keys
添加多个公钥到服务器下
cat id_rsa.pub >> ./ssh/authorized_keys

putty 产生的密钥和 openssh 不兼容,需要转换
ssh-keygen -i -f /root/.ssh/linden.pub>/root/.ssh/authorized_keys
多个公钥文件可以>>添加

另外当利用 public key 方式进行用户认证登录时，也会问密码，这个密码是 ssh-keygen 时输入的 passphrase，我的理解这个密码并没有通过网络传送，而是在本地验证，目的是来保护这个存在客户端的 private key 的，而通过 password 方式认证时，password 是加密后在网上传送的

凡是私钥， 如 host_rsa_key，host_dsa_key，$home/.ssh/id_rsa，id_dsa 的属性都必须是 600，而公钥 如 host_rsa_key.pub $/HOME/.ssh/id_rsa.pub 等属性必须是 644，sshd 才能正常工作，不符合，sshd 拒绝认证，这点 man 中并不清楚，我是错了好些次，在检查 log 时发现的

544 ☆sshd_config

/etc/ssh/sshd_config

Port 22
#ssh 服务监听端口号

Protocol 2,1
#协议顺序

ListenAddress 0.0.0.0
#绑定 ip 地址

PremintRootLogin Yes
#允许 root 登录

PermintEmptyPasswords No
#是否允许空密码用户登录

PasswordAuthentication yes
#是否使用口令认证方式,如使用公钥认证方式为 no

545 ☆SSH 的防火墙

iptables -A INPUT -i www.xxx.yyy.zzz -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i www.xxx.yyy.zzz -p udp --dport 22 -j ACCEPT
在防火墙为端口 22 打开缺口

如果你已经在防火墙上穿出个洞，并且安全壳依然不想连接，编辑位于/etc/hosts.allow 的文件在其中增加下面的一行.
.sshd: ALL

546 ☆启动停止 SSH 服务器

service sshd restart

547 ☆使用 SSH

ssh 服务器的 ip 或域名
第一次登录的时候,程序会把服务器的公钥缓存到用户主目录的.ssh 目录下的 know_host 文件里

548 ☆SSH 测试

尝试从服务器它自己的终端登陆到自身。如果它不能工作，那么在远端也将如此。
ssh www.xxx.yyy.zzz (ssh 后面的为你的服务器自己的 ip 地址。)
Enter passphrase for key '/home/user/.ssh/id_rsa': '密码' (输入你的密匙保护口令。)

如果一切都很正常，你将被询问过你的密匙保护口令以后取得像这样的信息:Last login:time and date 上次登陆的时间和日期。

如果第一次这样做(译补:每次使用 ssh 到以前未曾尝试连接到的主机的时候,know_list 列表中没有目标主机信息时候)你将看到这样的信息:永久地增加地址 www.xxx.yyy.zzz 到已知主机列表。
在这之后，你将得到一个新的壳的提示符并且可以像以往那样地工作。只不过现在它是一个安全壳的对话。

549 ☆SSH 问题

如果安全壳守护进程不能工作或者(连接)不可到达，你将获得这样的信息:ssh: Connect to www.xxx.yyy.zzz port 22: Connection refused.。

在一个服务器上使用两个以太网卡，安全壳守护进程将默认地在端口 22 监听每一块以太网卡。这通常是你所希望的。

如果你不能登陆进这其中的一个 IP 地址，试试另一个

如果本地的接口工作了，但是外部的接口不能工作，这个故障出现在你的防火墙上。

确保端口 22 是开启状态，并且准许 TCP 以及 UDP 协议地通行:
iptables -L

550 ☆SSH 日志文件

你可以从日志文件中监视你的安全壳守护进程(sshd)在做些什么。使用超级用户 root 运行终端，并且保证 tail 正在运行。
tail -f /var/log/messages

551 ☆SSH 远程登陆图形窗口

改/etc/ssh/sshd_config 里的一个参数
X11Forwarding yes

应该用 ssh -X 来进行 X11 转发。不需要设置什么 DISPLAY 变量。

用-X 时,不用设置 export DISPLAY
ssh -X username@hostname

除了设置那个 sshd_config 外，还有还得注意一下远程主机下的权限设置是否正确(不能有多余的写权限)，否则用密钥方式登录会失败，使用 X11 转发时，也会出错。
chmod 700 ~/.ssh
chmod 700 ~/.ssh/authorized_keys

进行转发时，要求本人必须用于 X11 环境下（也许不是必须）进行同用户的转发。也就是说使用 su 后再转发，还是不成的。

windows 下面用 ssh+exceed 的时候，需要在那边 set DISPLAY="YOUR IP":0.0 export DISPLAY

echo $DISPLAY 看一看是 localhost:0.0

552 ☆SSH 传送文件

sftp 和 scp 从连接到 sshd 服务器上后，不需要任何专用的守护进程。为了使用 sftp 和 scp 你必须插入以下两行在配置文件/etc/ssh/sshd_config 中:
Subsystem sftp /usr/libexec/openssh/sftp-server
在这些修改之后，你必须重新启动 sshd

然后你就可以使用 sftp 和 scp 连接到运行 sshd 的主机上了

现在你能使用安全文件传输程序(sftp)或者图形文件传输程序(gftp GUI)进行安全的文件传输了。这个(gftp GUI) 图形用户接口文件传输程序是被推荐使用的。

553☆sftp

Sftp
(sftp 安全文件传输)是一个类 ftp 的客户端程序，它能够被用来在网络中传输文件。
它并不使用 FTP 守护进程(ftpd 或 wu-ftpd)来进行连接，而是有意义地增强系统的安全性。
Sftp 使用在数据连接上使用 ssh2，所以文件的传输是尽可能地安全。

使用 sftp 代替 ftp 两个主要的的原因是:
1、Password 从不用明文传输，防止 sniffer(嗅探器)的攻击。
2、数据在传输时被加密，使用刺探和修改连接非常困难。

可以使用命令:
sftp username@host
host 可以为 ip
当 sftp2 准备好了来接受连接时，它将显示一个状态提示符 sftp>。

• quit:
从这个应用程序中退出。

• cd directory:
改变当前的远程工作目录。

• lcd directory:
改变当前的本地工作目录。

• ls [-l] [file ...]:
列出在远地服务器上的文件名。如果是目录，则列出目录的内容。(默认情况下，子目录并不被访问)。当命令行中指定了-l，文件与目录的权限，属主，大小和修改时间被列出。

• lls [-l] [file ...]:
与 ls 一样，但是是对于本地文件操作。

• get [file ...]:
从远程端传送指定的文件到本地端。目录内容被递归地复制。

• put [file ...]:
从本地端传送指定的文件到远地端。目录内容被递归地复制。

• mkdir dir (rmdir dir):
尝试建立或删除参数中指定的目录。

通配符对于 ls,lls,get 和 put 是支持的。如?,*

554 ☆scp

scp
(scp 安全性复制)被用来在网络上安全地复制文件。它替代了不安全的 rcp 命令。它使用 ssh2 来进行数据传送:它使用的确认方式和提供的安全性与 ssh2 一样。

在这种方式下文件 filename 被复制成相同的名字。通配符可以使用。

文件到远地的主机 host1 上的 remote_dir 目录中。使用 scp 你可以输入:
scp /本地路径/本地文件名 用户名@远端 IP 地址:/远端路径

scp local_dir/* myname@host1:remote_dir
从目录 local_dir 复制所有文件到主机 host1 的目录 remote_dir 命令:

scp myname@host1:remote_dir/filename .
复制文件 filename 从 host1 的目录 remote_dir 到本地目录。

scp 支持许多选项并且允许在两个远地系统之间复制文件:

scp myname@host1:remote_dir/filename myname@host2:another_dir
显然，使用 scp，你必须知道远程机器的确切目录，所以在实际上 sftp 经常被作为首选使用。(口令方式未成功,公钥方式成功了)

555 ☆文件传输脚本

SSH PUSH（从本地发送到远程）:

代码:
$ tar cvf - . | gzip -c -1 | ssh user@host "cat > remotefile.gz"
$ tar czvf - . | ssh user@host "cat > remote.tar.gz"
$ ssh target_address cat < localfile ">" remotefile
$ ssh target_address cat < localfile - ">" remotefile
$ cat localfile | ssh target_address cat ">" remotefile
$ cat localfile | ssh target_address cat - ">" remotefile
注： 以上命令行中的 大于号> 两边有双引号(" ")，小于号< 两边则没有。

$ dd if=localfile | ssh target_address dd of=remotefile

$ ssh target_address cat <localfile "|" dd of=remotefile
$ ssh target_address cat - <localfile "|" dd of=remotefile
$ (cd SOURCEDIR && tar cf - .) | ssh target_address "(cd DESTDIR && tar xvpf -)"
$ (cd SOURCEDIR && tar cvf - .) | ssh target_address "(cd DESTDIR && cat - > remotefile.tar)"
$ (cd SOURCEDIR && tar czvf - .) | ssh target_address "(cd DESTDIR && cat - > remotefile.tgz)"
$ (cd SOURCEDIR && tar cvf - . | gzip -1 -) | ssh target_address "(cd DESTDIR && cat - > remotefile.tgz)"
$ ssh target_address "(nc -l -p 9210 > remotefile &)" && cat source-file | gzip -1 - | nc target_address 9210
$ cat localfile | gzip -1 - | ssh target_address cat ">" remotefile.gz

SSH PULL（从远程获取到本地）:

代码:
$ ssh target_address cat remotefile > localfile
$ ssh target_address dd if=remotefile | dd of=localfile
$ ssh target_address cat "<" remotefile > localfile
$ ssh target_address cat "<" remotefile.gz | gunzip >localfile

SSH COMPARE（本地和远程文件比较）:

代码:
用远程计算机 CPU 执行比较指令
$ ssh target_address cat remotefile | diff - localfile
$ cat localfile | ssh target_address diff - remotefile

用本地计算机 CPU 执行比较指令
$ ssh target_address cat <localfile "|" diff - remotefile

注：
1、把文件压缩后再传送速度一般要比直接传送更快。
2、以上命令行中 'localfile' 和 ' remotefile' 可以是文件、目录、镜像、硬盘分区、或者硬盘驱动器(files, directories, images, hard drive partitions, or hard drives)。

FTP VIEW（从本地查看远程文件）:

代码:
$ ftp> get file.gif "| xv -"
$ ftp> get README "| more"

FTP PUSH（从本地发送到远程）:

代码:
$ ftp> put "| tar cvf - ." myfile.tar
$ ftp> put "| tar cvf - . | gzip " myfile.tar.gz

FTP PULL（从远程获取到本地）:

代码:
$ ftp> get myfile.tar "| tar xvf -"

556 ☆ssh-agent

ssh-agent 程序如同一个看门人，它根据需要安全地提供对安全密钥(不能用于口令方式)的访问。ssh-agent 启动后，它就会在后台运行，并且可以由 ssh 和 scp 程序等其他 OpenSSH 应用程序所使用。这就使得 ssh 程序可以请求一个已经解密了的密钥，而不是在每次需要时向您询问私钥的安全口令。

[root@askpanda ~]# ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-sArQMR4389/agent.4389; export SSH_AUTH_SOCK;
SSH_AGENT_PID=4390; export SSH_AGENT_PID;
echo Agent pid 4390;

我们可以使用 shell 的 eval 命令来让 shell 执行 ssh-agent 显示的输出命令：(每个用户下执行)
[root@askpanda ~]# eval `ssh-agent`
Agent pid 4510

现在我们就已经可以使用 ssh-agent 共享我们的口令
[root@askpanda ~]# ssh-add (每个用户下执行)
Enter passphrase for /home/panda/.ssh/id_dsa: (enter passphrase)
Identity added: /home/panda/.ssh/id_dsa
(/home/panda/.ssh/id_dsa)

现在，当我们访问 panda 时，不会再被提示输入口令：
[root@askpanda ~]# ssh root@panda
[root@panda ~]# exit

557 ☆xhost

在本地主机上的任意一个 xterm 中执行 xhost,用来允许远程的其它主机可以和本地主机的 X server 联网:

xhost + 远程访问 ip 或计算机名

如果不指定任何 ip 地址,则表示权限完全放开,这会带来安全问题,要小心!

确认本地主机的 xfs 是运行的.用 ps 检查一下进程.

从本地主机(192.168.152.128)上通过网络登录到远程主机 192.168.152.129 上,你用 telnet,ssh,rsh 都可以.设置 DISPLAY 变量.
export DISPLAY=192.168.152.128:0

客户端用 xterm

语法
xhost Show current security settings
xhost + Disable security; allow connections from any
system
xhost - Enable security
xhost +apps.xyz.com Allow connection from apps.xyz.com
xhost -apps.xyz.com Disable connections from apps.xyz.com

You can send the display of an X Client program to a remote system by using the -display
option.
xterm -display localhost:0.0 &

558 ☆VNC

VNC 就是 vitual network computing 的缩写,它支持许多操作平台,甚至可在浏览器中操作.

vnc client 通过架构在 tcp/ip 上的 vnc 协议与 vnc server 沟通,通过认证后,把 X server 的桌面环境,输入设备,和 X 资源交给 vncserver 掌控,vnc server 将桌面环境通过 vnc 协议送给 vnc client 端.让 vnc client 来操纵 vnc server 桌面环境和输入设备.

559 ☆VNC 安装

vnc-server

560 ☆VNC server 密码

vncpasswd

密码保存在~user/.vnc/passwd

561 ☆VNC server 启动

vncserver [:桌面号]
每个用户连接需要占用一个桌面,可以同时连接到一个桌面号

vncserver -kill [:桌面号]
关闭一个桌面号

注意运行后显示的信息,记下所用的端口号,一般从 1 开始,因为 0 被 x server 占用了.现在,你就能提供 vnc 服务了.

562 ☆vncservers

/etc/sysconfig/vncservers
会在启动的时候自动启动桌面号
VNCSERVERS=”桌面号:使用的用户名”

VNCSERVERS="1:root"
VNCSERVERARGS[1]="-geometry 800x600"

563 ☆vnc 防火墙

VNC 服务使用的 TCP 端口从 5900 开始
基于 Java 的 VNC 客户程序 Web 服务 TCP 端口从 5800 开始

iptables -I INPUT -p tcp --dport 5901 -j ACCEPT
iptables -I INPUT -p tcp --dport 5801 -j ACCEPT

564 ☆vncviewer

vnc server 启动成功后,你就可用 vncviewer 来远程控制桌面了.
vncviewer xxx.xxx.xxx.xxx:display number

vncviewer 192.168.152.129:1
按要求输入密码就可以看到远程的桌面了.

注意:viewers 需要在 16 位色的显示模式下工作，如果您的操作系统中没上 16 位色，那么请您及时的调整您计算机的显示模式。不然 vncviewer 无法正常工作。

565 ☆xstartup

修改$HOME/.vnc/xstartup 这个文件.

把所有内容的行前加上#(可选),再在接尾部
最后一行把 twm &改成
startkde & 或者 gnome-session &
你当然可用你喜好的桌面代替.我这是用 kde 来代替 twm,速度会慢少少,但用起来方便不少.

注意要重新启动 vnc server

566 ☆通过浏览器使用 vnc

通过浏览器使用 vnc,要注意端口号的变化.

假设 vnc server 是 192.168.152.129:1 的话,那么,可用浏览器访问 http://192.168.152.129:5801
端口号=display number + 5800

567 ☆x0vncserver

x0vncserver -PasswordFile=/当前用户主目录/.vnc/passwd

运行后,远程可以连接到:0 桌面,直接操作用户的桌面,实现远程协助

568 ☆Apache 启动

service httpd restart

重新启动 httpd:
service httpd reload

手册在 localhost/manual 上浏览

569 ☆Apache 防火墙

iptables -I INPUT -p tcp --dport 80 -j ACCEPT

570 ☆Apache 配置文件

/etc/httpd 下有
drwxr-xr-x 7 root root 4096 7 月 6 18:07 conf
drwxr-xr-x 2 root root 4096 6 月 1 14:55 conf.d
lrwxrwxrwx 1 root root 19 6 月 1 14:54 logs -> ../../var/log/httpd
lrwxrwxrwx 1 root root 27 6 月 1 14:54 modules -> ../../usr/lib/httpd/modules #所有模块
lrwxrwxrwx 1 root root 13 6 月 1 14:54 run -> ../../var/run

571☆httpd.conf

/etc/httpd/conf/httpd.conf
#这是 Apache 服务器主要配置文件。
#它包含服务器的影响服务器运行的配置指令。
#
#这些配置指令被分为下面三个部分:
#1. 控制整个 Apache 服务器行为的部分(即全局环境变量);这里设置的参数将影响整个 Apache 服务器的行为;
#2. 定义主要或者默认服务参数的指令，也为所有虚拟主机提供默认的设置参数
#3. 虚拟主机的设置参数
#
#配置和日志文件名:如果你指定的文件名以"/"开始(win32 下以"dirver:/"),服务器将使用绝对路径，
#如果文件名不是以"/"开始的，那么它将把 ServerRoot 的值附加在文件名的前面，
#例如，对"logs/foo.log"，如果 ServerRoot 的值为"/etc/httpd/"，则该文件应为"/etc/httpd/logs/foo.log"
#

第一区:全局环境变量

ServerType [standalone|inetd]
这个配置选项指定如何运行 WEB 服务器。
Apache 可以使用两种方法来运行服务器:standalone(独立的)和 inetd(由 inetd 运行的)。
standalone 表示 WEB 服务进程以一个单独的守候进程的方式在后台侦听是否有客户端的请求，如果有就生成一个子进程来为其服务。在 standalone 模式下，不存在对每个请求启动新进程的开销，所以它的效率更高;而 inetd 模式被认为比 standalone 模式更具安全性。此种模式下，WWW 服务器侦听特定端口的连接请求。当客户机发出到特定端口地址的连接请求时，主服务器进程启动子 WWW 服务进程来服务该请求。另外还需要告诉主服务器进程侦听的特定端口地址，使用命令:Port 80
inetd 参数表示 WEB 服务不是以一个单独的守候进程的形式支持。而是由 xinetd 这个超级服务器守候进程进行代劳，当它收到一个客户端的 WEB 服务请求的时候，再启动一个 WEB 服务进程为其服务。xinetd 是侦听有小于 1024 的端口连接请求的 Internet 守护进程(一个服务器进程)。与前面的方法不同，当客户系统发出到 WWW 服务器的连接请求时，xinetd 启动一个 WWW 服务器进程，由此进程服务此请求，完成服务后即退出。如果选择通过 xinetd 服务器来运行 Apache，需要编辑/etc/xinetd.conf 文件为 Apache 添加一条新的记录:httpd stream tcp nowait httpd /etc/httpd/bin/httpd -f /etc/httpd/conf/httpd.conf
修改了/etc/xinetd.conf 文件后，就需要修改/etc/services 中添加一行
httpd 80/tcp httpd
做完以上修改后，需要重新启动 xinetd 进程。首先，使用以下命令取得 xinetd 的进程 ID:
ps auxw |grep xinetd
然后执行命令:kill -HUP <inetd 的进程 ID>
从功能的角度看，这两种方法几乎是相同的。但它们之间实际有很大区别，区别在于服务器的性能。一个由 xinetd 运行的服务器进程在它结束对请求服务的同时立刻退出。而在 standalone 模式

下，子 WWW 服务器进程在退出之前要挂起一段时间，这就给它们提供了机会，可以重新用来服务新的请求。

ServerRoot [fully qualified path name]
它指出服务器保存其配置、出错和日志文件等的根目录。注意！如果你想要将它指定为 NFS 或其它网络上的位置，请一定要去阅读与 LockFile 有关的文档
例:ServerRoot "/etc/httpd"
路径的结尾不要添加斜线。

#串行访问的锁文件必须保存在本地磁盘上
<IfModule !mpm_winnt.c>
<IfModule !mpm_neware.c>
#LockFile logs/accept.lock
</IfModule>
</IfModule>

#ScoreBoardFile:用来保存内部服务进程信息的文件。
#如果未指明(默认)，记分板(scoreboard)将被保存在一个匿名的共享内存段中，并且它不能被第三方软件所使用。
#如果指定了，要确保不能使用两个 Apache 使用同一个记分板文件，
#这个记分板文件必须保存在本地磁盘上。
<IfModule !mpm_netware.c>
<IfModule !perchild.c>
#ScoreBoardFile logs/apache_runtime_status
<IfModule>
<IfModule>

#PidFile [Path]:记录服务器启动进程号的文件。
<IfModule !mpm_neware.c>
PidFile run/httpd.pid
</IfModule>

Timeout [second]
接收和发送前超时秒数,只要客户端超过这里设定的秒数还没有完成一个请求的话，服务端将终止这次请求服务。如果网络速度较慢的话，建议在此设置较大的数值。以给客户端更多机会。
例:Timeout 120

#KeepAlive:是否允许稳固的连接(每个连接有多个请求)，
#设为"Off"则停用。On
KeepAlive Off

#MaxKeepAliveRequests:在稳固连接期间允许的最大请求数，

#设为 0 表示无限制接入。
#我们推荐你将其设为一个较大的值，以便提高性能
MaxKeepAliveRequests 100

#KeepAliveTimeout:在同一个连接上从同一台客户上接收请求的秒数
KeepAliveTimeout 15

##Server-Pool 大小设定(针对 MPM 的)

prefork MPM
StartServers:启动时服务器启动的进程数
MinSpareServers:保有的备用进程的最小数目
MaxSpareServers:保有的备用进程的最大数目
MaxClients:服务器允许启动的最大进程数
MaxRequestsPerChild:一个服务进程允许的最大请求数
<IfModule prefork.c>
StartServers 5
MinSpareServers 5
MaxSpareServers 10
MaxClients 150
MaxRequestPerChild 0
</IfModule>

worker MPM
StartServers:服务器启动时的服务进程数目
MaxClients:允许同时连接的最大用户数目
MinSpareThreads:保有的最小工作线程数目
MaxSpareThreads:允许保有的最大工作线程数目
ThreadsPerChild:每个服务进程中的工作线程常数
MaxRequestsPerChild:服务进程中允许的最大请求数目
<IfModule worker.c>
StartServers 2
MaxClients 150
MinSpareThreads 25
MaxSpareThreads 75
ThreadsPerChild 25
MaxRequestsPerChild 0
</IfModule>

perchild MPM
NumServers:服务进程数量
StartThreads:每个服务进程中的起始线程数量

```
# MinSpareThreads:保有的最小线程数量
# MaxSpareThreads:保有的最大线程数量
# MaxThreadsPerChild:每个服务进程允许的最大线程数
# MaxRequestsPerChild:每个服务进程允许连接的最大数量
<IfModule perchild.c>
NumServers 5
StartThreads 5
MinSpareThreads 5
MaxSpareThreads 10
MaxThreadsPerChild 20
MaxRequestsPerChild 0
</IfModule>

# WinNT MPM
# ThreadsPerChild:服务进程中工作线程常数
# MaxRequestsPerChild:服务进程允许的最大请求数
<IfModule mpm_winnt.c>
ThreadsPerChild 250
MaxRequestsPerChild 0
</IfModule>

# BeOS MPM
# StartThreads:服务器启动时启动的线程数
# MaxClients:可以启动的最大线程数(一个线程等于一个用户)
# MaxRequestsPerThread:每个线程允许的最大请求数
<IfModule beos.c>
StartThreads 10
MaxClients 50
MaxRequestsPerThread 10000
</IfModule>

# NetWare MPM
# ThreadStachSize:为每个工作线程分配的堆栈尺寸
# StartThreads:服务器启动时启动的线程数
# MinSpareThreads:用于处理实发请求的空闲线程数
# MaxSpareThreads:空闲线程的最大数量
# MaxThreads:在同一时间活动的最大线程数
# MaxRequestPerChild:一个线程服务请求的最大数量，
#  推荐将其设置为 0，以实现无限制的接入
<IfModule mpm_netware.c>
ThreadStackSize 65536
StartThreads 250
```

MinSpareThreads 25
MaxSpareThreads 250
MaxThreads 1000
MaxRequestPerChild 0
</IfModule>

OS/2 MPM
StartServers:启动的服务进程数量
MinSpareThreads:每个进程允许的最小空闲线程
MaxSpareThreads:每个进程允许的最大空闲线程
MaxRequestsPerChild:每个服务进程允许的最大连接数
<IfModule mpmt_os2.c>
StartServers 2
MinSpareThreads 5
MaxSpareThreads 10
MaxRequestsPerChild 0
</IfModule>

Listen:允许你绑定 Apache 服务到指定的 IP 地址和端口上，以取代默认值
使用如下命令使 Apache 只在指定的 IP 地址上监听，可多个 Listen
以防止它在 IP 地址 0.0.0.0 上监听
Listen 12.34.56.78:80
Listen 80

动态共享支持(DSO)
为了能够使用那些以 DSO 模式编译的模块中的函数，你必须有相应的"LoadModule"行，
因此，在这里包含了这些指令，以便能在使用它之前激活。
那些静态编译的模块不需要在这里列出 (即以"httpd -l"列出的模块)
示例:
LoadModule foo_module modules/mod_foo.so

ExtendedStatus:当调用"server-status"时，控制 Apache 是产生"全"状态
信息(ExtendedStatus On)，还是产生基本信息(ExtendedStatus Off)。
默认为 off
ExtendedStatus On

SetEnvIf <Remote_Addr|Remote_host|Remote_user|Request_Method|Request_protocol|Request_URL>
"condition" VarName
只要检查属性中符合 Condition，就定义名字为 VarName 的变量
condition 为 unix 中的正则表达式(Regular Expression)，可以查看 man regex 获取详细资料
SentEnvIf mod_setenvif.c 提供<默认>

ResourceConfig
#资源文件所在,让 srm.conf 为空，则 httpd.conf 的相关值生效

AccessConfig
#访问权限文件的所在路径

LoadModule
#加载模块

AddModule
#加载模块的顺序

HeaderName
#页面题头文件

ReadmeName
#页面页尾文件

IndexIgnore
#隐藏文件规则

IndexOptions
#控制目录列表的外观

Alias /path/ "/<path>/"
#别名定义

第二区:"主"服务配置

这一区建立被 "主" 服务器用的指令值,以回应那些不被 <VirtualHost>定义处理的任何请求。
这些数值也提供默认值给后面定义的<VirtualHost>容器。
如果<VirtualHost>中有定义，那么这里定义的指令值将被<VirtualHost>中的定义所覆盖。

<IfModule !mpm_winnt.c>
<IfModule !mpm_neware.c>
如果你想使 httpd 以另外的用户或组来运行，你必须在开始时以 root 方式启动,然后再将它切换为你想要使用的用户或组。
User/Group:运行 httpd 的用户和组
在 SCO (ODT3)上使用"User nouser"和"Group nogroup"
在 HPUX 上，你可能不能以 nobody 身份使用共享内存，建议创建一个 www 用户。
注意一些核心(kernel)在组 ID 大于 60000 时拒绝 setgid(Group)或 semctl(IPC_SET)，节在这些系统上不要使用"Group #-1"。

User nobody
Group #-1
</IfModule>
</IfModule>

ServerAdmin [you E-Mail address]:你的邮件地址，当发生问题时 Apache 将向你发出邮件。
作为一个出错文档，这个地址显示在 server-generated 页上，
#管理员信箱命令，用来设置 WEB 管理员的 E-Mail 地址。这个地址会出现在系统连接出错的时候，以便访问者能够将情况及时地告知 WEB 管理员。
ServerAdmin root@panda.com

ServerName 指定 Apache 用于识别自身的名字和端口号。可以为 ip 地址
通常这个值是自动指定的，但是我们推荐你显式的指定它以防止启动时出错
如果你为你的主机指定了一个无效的 DNS 名，server-generated 重定向将不能工作。
参见 UseCanonicalName 指令
如果你的主机没有注册 DNS 名，在这里键入它的 IP 地址
无论如何，你必须使用它的 IP 地址来提供服务，
这里使用一种容易理解的方式重定向服务
ServerName www.dalouis.com:80

UseCanonicalName:决定 Apache 如何构造 URLS 和 SERVER_NAME 和 SERVER_PORT 的指令。
当设置为 "Off"时,Apache 会使用用户端提供的主机名和端口号。
当设置为"On",Apache 会使用 ServerName 指令的值。
UseCanonicalName Off

DocumentRoot [Path]:你的文档的根目录。默认情况下，所有的请求从这个目录进行应答。
但是可以使用符号链接和别名来指向到其他的位置。
DocumentRoot "/home/redhat/public_html"

Apache 可以存取的每个目录都可以配置存取权限(包括它的子目录)。
首先，我们配置一个高限制的特征。
这将禁止访问文件系统所在的目录，并添加你希望允许访问的目录块。
如下所示
<Directory />
Order Deny,Allow
Deny from all
</Directory>

注意从这里开始你一定要明确地允许哪些特别的特征能够被使用。所以，如果 Apache 没有象你所期待的那样工作的话,请检查你是否在下面明确的指定它可用。

这将改变到你设置的 DocumentRoot
<Directory "/home/redhat/public_html">
Options:这个指令的值可以是"None"，"All"，或者下列选项的任意组合:
Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
注意，"MultiViews"必须被显式的指定，"Options All"不能为你提供这个特性。
Options FollowSymLinks

Option 的参数:

All	准许以下所有功能(MultiViews 除外);
MultiViews	请求 a 的时候,不存在,返回 a.*文件,准许内容协商的 Multiviews;
Indexes	允许目录浏览.若该目录下无 index 文件,则准许显示该目录下的文件以供选择;
IncludesNOEXEC	准许 SSI(Server-side Includes),但不可使用#exec 和#include 功能;
Includes	准许 SSI;
FollowSymLinks	准许符号链接到其他目录;
ExecCGI	准许该目录下可以使用 CGI。

AllowOverride 控制那些被放置在.htaccess 文件中的指令。
它可以是"All"，"None"，或者下列指令的组合:
Options FileInfo AuthConfig Limit

AllowOverride None

AllowOverride [options]
命令则是用来决定是否准许在"httpd.conf"文件中设定的权限是否可以被在文件".htaccess"中设定的权限覆盖。
All 允许.htaccess 覆盖 httpd.conf 中的所有目录设置
Authconfig 允许覆盖认证指令 AuthName AuthType AuthUserFile AuthGroupFile 等
FileInfo 允许覆盖目录存取文件设置 AddEncodeing AddLanguage AddType CookieTracking CookieName 等
Indexes 允许覆盖目录索引设置
Limit 允许覆盖 allow deny order require
None 不允许覆盖 httpd.conf 中的所有目录设置
Options 允许覆盖 Options 的设置

控制谁可以获得服务。
Order allow,deny
Allow from all

Order 命令:用来设定谁能从这个服务器取得控制。它也有两个参数:
allow 可以取得控制;
deny 禁止取得控制。

</Directory>

UserDir [Path|disable]:指定在得到一个~user 请求时将会添加到用户 home 目录后的目录名。
UserDir public_html
用来指定个人主页的位置。如果你有一个用户 panda，那么它主目录是"/home/panda"，当客户端输入"http://yourdomain/~panda"，系统就会到对应的目录"/home/panda/UserDir/"中去寻找。其中"UserDir"就是在 UserDir 命令中设置的指定目录。

为防止在 UserDir 指令上的漏洞，对 root 用户设置象"./"这样的 UserDir 是非常有用的。
如果你使用 Apache 1.3 或以上版本，我们强烈建议你在你的服务器配置文件中包含下面的行
UserDir disabled root

DirectoryIndex [filename]:用来声明首页文件名称。定义请求是一个目录时，Apache 向用户提供服务的文件名,用空格分开,按顺序查找文件
index.html.var 文件(一个类型映象文件)用于提供一个文档处理列表，
出于同样的目的，也可以使用 MultiViews 选项，但是它会非常慢。
DirectoryIndex index.php index.html index.html.var

AccessFileName:在每个目录中查询为目录提供附加配置指令的文件的文件名。
参见 AllowOverride 指令。

```
AccessFileName .htaccess

# 下面的行防止.htaccess 和.htpasswd 文件被 Web 客户查看。
<Files ~ "^\.ht">
Order allow,deny
Deny from all
</Files>

# Typeconfig:定义在哪里查询 mime.types 文件。
TypeConfig conf/mime.types

# DefaultType:定义当不能确定 MIME 类型时服务器提供的默认 MIME 类型。
# 如果你的服务主要包含 text 或 HTML 文档，"text/plain"是一个好的选择;
# 如果大多是二进制文档，诸如软件或图像，你应使用
# "application/octer-stream"来防止浏览器象显示文本那样显示二进制文件。
DefaultType text/plain

# mod_mime_magic 允许服务器从自己定义自己类型的文件中使用不同的线索(hints)，
# 这个 MIMEMagicFile 指令定义 hints 定义所在的文件。
<IfModule mod_mime_magic.c>
MIMEMagicFile conf/magic
</IfModule>

# HostnameLookups:指定记录用户端的名字还是 IP 地址，例如，本指令为 on 时
# 记录主机名，如 http://www.apache.org;为 off 时记....62.129.132。
# 默认值为 off，这要比设为 on 好得多，因为如果设为 on 则每个用户端请求都将会
# 至少造成对 nameserver 进行一次查询。
HostnameLookups Off

# EnableMMAP:控制是否进行内存转储(如果操作系统支持的话)。
# 默认为 on，如果你的服务器安装在网络文件系统上(NFS)，请关闭它。
# 在一些系统上，关闭它会提升系统性能(与文件系统类型无关);
# 具体情况请参阅 http://httpd.apache.org/docs-2.0/mod...tml#enablemmap
# EnableMMAP off

# EnableSendfile:控制是否使用 sendfile kernel 支持发送文件
# (如果操作系统支持的话)。默认为 on，如果你的服务器安装在网络文件系统
# (NFS)上，请你关闭它。
# 参见 http://httpd.apache.org/docs-2.0/mod...enablesendfile
# EnableSendfile off

# ErrorLog [log filename]:错误日志文件定位。
```

如果你没有在<VirtualHost>内定义 ErrorLog 指令，这个虚拟主机的错误信息
将记录在这里。如果你在那儿定义了 ErrorLog，这些错误信息将记录在你所
定义的文件里，而不是这儿定义的文件。
ErrorLog <path|pipe|syslog>
日志发送到文件:ErrorLog /var/www/log
日志发送到 syslog:ErrorLog syslog:local <1-7> 默认为 local 7
日志发送给管道:ErrorLog "|/usr/local/apache/bin/errordb insert"
ErrorLog logs/error_log

LogLevel:控制记录在错误日志文件中的日志信息数量。
可能的值包括:debug，info，notice，warn，error，crit，alert，emerg。
LogLevel <emerg|alert|crit|warn|notice|info|debug> 简略-->详细 严重-->轻微
LogLevel warn

下面的指令为 CustomLog 指令定义格式别名。
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
通用日志的默认格式等价于
LogFormat "%h %l %u %t \" %r \" %>s %b"
LogFormat "%401u" #记录未认证的用户名
apache 预先定义了四种日志的格式:<combined|common|referer|agent>

LogFormat:自定义日志格式
%a client ip
%A server ip
%b 响应 http 标头以外的字节数
%{eviron}e 以 eviron 命名的变量的值
%f 被请求文件的路径
%h client hostname
%{reqheader}i 名字为 reheader 的标头的值
%l client 的系统环境，IdentifyCheck 为 On 且内容协商成功
%{modnote}n 名字为 modnote 的短信的值
%p server port
%P server PID
%r 请求的第一行，包括 URLmodth
%s 原始响应信息
%>s 返回浏览器的响应状态
%t 请求的发生时间[18/Nov?1999:13:02:58-0600]
%{timefmt}t 定义时间的显示格式，因为基本不需要修改，所以省略了...

%T 处理请求的秒数
%u 请求用户的登陆名
%U 被请求的 URL，不包括参数
%v server name
%V 根据 UseCanonicalName 得到服务器名称

你需要安装了 mod_logio.c 模块才能使用%I 和%O。
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio

指定接入日志文件的定位和格式(一般日志格式)。
如果你没有在<VirtualHost>内定义这个指令，传输信息将记录在这里，
如果你定义了这个指令，则记录在你指定的位置，而不是这儿定义的位置。
CustomLog logs/access_log common

如果你想要记录 agent 和 referer 信息，可以使用下面的指令
CustomLog logs/referer_log referer
CustomLog logs/agent_log agent

如果你想要使用一个文件记录 access，agent 和 referer 信息，
你可以如下定义这个指令:
CustomLog logs/access_log combined

CoustomLog "/var/logs/httpd/access_logs" env = myCheck #否定为 env =! myCheck
LogFormat CustomLog 由 mod_log_config.c 提供<默认>
在缺省情况下，它用通用日志格式 CLF 规范来写。CLF 日志文件内对每个请求均有一个单独行，
形如:host ident anthuser date request status bytes
其含义如下:
host 客户端主机的全称域名或 IP 地址;
ident 存放客户端报告的识别信息;
authuser 如果是基于用户名认证的话，值为用户名;
date 请求的日期与时间;
request 客户端的请求行;
status 返回到客户端的三们数字的 HTTP 状态码;
bytes 除去 HTTP 头标外，返回给客户端的字节数。

分析日志文件
列出访问过本网站的主机名或 IP:
cat /var/log/httpd/access_log | awk '{print $1}'
使得列出的表中，每个主机只出现一次。我们可以使用:
cat /var/log/httpd/access_log | awk '{print $1}' | wc -l

ServerTokens

这个指令定义包含在 HTTP 回应头中的信息类型。默认为"Full"，
这表示在回应头中将包含模块中的操作系统类型和编译信息。
可以设为列各值中的一个:
Full | OS | Minor | Minimal | Major | Prod
Full 传达的信息最多，而 Prod 最少。
ServerTokens Full

随意的添加包含服务器版本和虚拟主机名字一行信息到 server-generated 输出页中
(内部错误文档，FTP 目录列表，mod_status 和 mod_info 输出等等，除了 CGI 错误
或自定义的错误文档以外)。
设为"EMail"将包含一个指向 ServerAdmin 的 mailto:连接。
可以为如下值:On | Off | EMail
ServerSignature On

Aliases:在这时添加你需要的别名，格式如下:
Alias 别名 真实名
注意，如果你在别名的末尾包含了"/"，那么在 URL 中也需要包含"/"。
因此，"/icons"不是这个示例中的别名。
如果别名中以"/"结尾，那么真实名也必须以"/"结尾，
如果别名中省略了结尾的"/"，那么真实名也必须省略。
我们使用别名"/icons/"来表示 FancyIndexed 目录列表，如果你不使用、
FancyIndexing，你可以注释掉它。
用来创建虚拟目录
Alias /icons/ "/usr/local/apache2/icons/"

<Directory "/usr/local/apache2/icons">
Options Indexes MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>

#
这将改变 ServerRoot/manual。这个别名提供了手册页所在的位置，
即使你改变了你的 DocumentRoot。如果你对有无手册页并不在意的话，
你可以注释掉它。
#
Alias /manual "/usr/loacl/apache2/manual"

<Directory "/usr/local/apache2/manual">
Options Indexes FollowSymLinks MultiViews IncludesNoExec
AddOutputFilter Includes html

Order allow,deny
Allow from all
</Directory>

ScriptAlias:指定包含服务脚本的目录。为脚本程序目录起个别名
ScriptAliases 本质上与 Aliases 一样，除了这里的文档在请求时做为程序处理处理以外。
尾部的"/"规则与 Alias 一样
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

要为 CGI 目录设置合适的权限，一般是只允许 Apache 有读取和执行的权限但没有写的权限。
<Directory>
Options ExecCGI #以允许使用 CGI
SetHandler cgi-script #设定 CGI 脚本类型
</Directory>
如果要考虑提高安全性的话，应将集中的 CGI 程序目录保存在 DocumnetRoot 目录外，使得访问者不能直接访问 CGI 程序。

这里是添加 php 4 支持的指令
AddType application/x-httpd-php .php
LoadModule php4_module modules/libphp4.so

<IfModule mod_cgid.c>
#
添加 mod_cgid.c 设置，mod_cgid 提供使用 cgid 进行通讯的 UNIX 套接字的
脚本接口路径。
#
Scriptsock logs/cgisock
</IfModule>

#
将"/usr/local/apache2/cgi-bin"改为你的 ScriptAliased 指定的 CGI 目录，
如果你配置了的话。
#
<Directory "/usr/local/apache2/cgi-bin">
AllowOverride None
Options None
Order allow,deny
Allow from all
</Directory>

#

```
# Redirect 允许你告诉客户端使用存在于服务器名字空间中的文档，
#  而不是现在的，这帮助客户定位那些改变了位置的文档。
#  例如:
# Redirect permanent /foo http://www.panda.com/bar

#
#  控制 server-generated 目录列表显示的指令
#

#
# IndexOptions:控制 server-generated 目录列表显示特征。
#
IndexOptions FancyIndexing VersionSort

#
# AddIcon*  指令告诉服务器不同扩展名的图象文件如何显示，
#  只适用于 FancyIndexed 指令
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
```

```
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

#
# DefaultIcon  为那些没有显式定义图标的文件提供处理
#
DefaultIcon /icons/unknown.gif

#
# AddDescription 允许你在 server-generated 索引后放置一个简短的说明。
#  只对 FancyIndexed 指令有效。
#  格式:AddDescription "说明"  文件名
#
# AddDescription "GZIP compressed document" .gz
# AddDescription "tar archive" .tar
# AddDescription "GZIP compressed tar archive" .tgz

#
# ReadmeName 指定服务器默认查找的 README 文件的名字，并添加到目录列表中
#
# HeaderName 指定目录列表前缀文件的文件名
ReadmeName README.html
HeaderName HEADER.html

#
# IndexIgnore 指定目录索引忽略并且不包含在列表中的文件名集合，
#  支持 shell 类型的通配符。
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

#
# AddEncoding 允许你在信息传送中使用(Mosaic/X 2.1+)解压缩信息，
#  注意:不是所有的浏览器都支持这个选项。
#  尽管名字相似，但是下列的指令与上面的 FancyIndexing 定制指令不同。
#
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz

#
# DefaultLanguage 和 AddLanguage 允许你指定文档的语言。
#  这使你可以让用户用容易理解的语言浏览文档。
```

#
指定默认的语言，这意味着所有没有指定语言的包都将使用该语言。
多数情况下，你也许并不想设置它，除非你确信这样做是正确的。
通常，不使用确定的语言比使用错误的语言要好。
#
DefaultLanguage nl
#
注意 1:作为语言关键字的词缀毫无疑问是不能一样的--采用波兰
文的文档(网络标准语言代码是 pl)将希望使用"AddLanguage pl .po"
来避免与 perl 脚本的一般词缀产生二义性。
#
注意 2: 下面的例子举例说明在一些范例中语言的二字符缩写与它的国家
的二字符缩写不相同,例如 "Danmark/dk" 和 "Danmark/da" 的比较.
#
注意 3: 在 "ltz" 的情况下我们使用三字符词缀，违犯了 RFC 的规定，
运行中将修复它并使用 RFC1766 标准取得参考数据。
#
Danish (da) - Dutch (nl) - English (en) - Estonian (et)
French (fr) - German (de) - Greek-Modern (el)
Italian (it) - Norwegian (no) - Norwegian Nynorsk (nn) - Korean (ko)
Portugese (pt) - Luxembourgeois* (ltz)
Spanish (es) - Swedish (sv) - Catalan (ca) - Czech(cz)
Polish (pl) - Brazilian Portuguese (pt-br) - Japanese (ja)
Russian (ru) - Croatian (hr)
#
AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .et
AddLanguage fr .fr
AddLanguage de .de
AddLanguage he .he
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
AddLanguage pl .po
AddLanguage ko .ko
AddLanguage pt .pt
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pt-br .pt-br
AddLanguage ltz .ltz

```
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .sv
AddLanguage cz .cz
AddLanguage ru .ru
AddLanguage tw .tw
AddLanguage zh-tw .tw
AddLanguage hr .hr

# LanguagePriority 允许你在会话过程中优先使用一些语言。
#
#  以优先次序递减的方式列出它们。我们或多或少地采用按字母排列顺序的方式
#  排列它们。也许你想要改变这个顺序。
LanguagePriority en da nl et fr de el it ja ko no pl pt pt-br ltz ca es sv tw

#
# ForceLanguagePriority  允许你为 MULTIPLE CHOICES(Prefer)[在通讯的情况下]
#  或 NOT ACCEPTABLE(Fallback)[没有可接受的语言匹配的情况]提供一个结果页。
#
ForceLanguagePriority Prefer Fallback

#
#  为发送出的所有页指定默认的字符集，这总是一个好主意，并且为你的
# web 站点的国际化打开了大门，这不正是你曾经想要的吗。同样地，指定
#  默认字符集有一些小的损害，如一个使用 iso-8859-1(latin1)标准命令
#  的页面，除非以别的方式指定例如你仅仅以显式方式声明它。
#  也有一些与那些总是鼓励你使用默认字符集的 javascropt 和 URL 语法有关
#  的浏览器安全原因。
#
#AddDefaultCharset ISO-8859-1
AddDefaultCharse GB2312

#
#  一般以文件扩展名的方式使用字符集。也许你想要避免与语言扩展发生
#  碰撞，除非你在每次改变后都做了很好的测试。
#  参见 http://www.iana.org/assignments/char...??得字符集
#  的名字列表和它们各自的 RFCs。
#
AddCharset ISO-8859-1 .iso8859-1 .latin1
AddCharset ISO-8859-2 .iso8859-2 .latin2 .cen
AddCharset ISO-8859-3 .iso8859-3 .latin3
AddCharset ISO-8859-4 .iso8859-4 .latin4
```

```
AddCharset ISO-8859-5 .iso8859-5 .latin5 .cyr .iso-ru
AddCharset ISO-8859-6 .iso8859-6 .latin6 .arb
AddCharset ISO-8859-7 .iso8859-7 .latin7 .grk
AddCharset ISO-8859-8 .iso8859-8 .latin8 .heb
AddCharset ISO-8859-9 .iso8859-9 .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5 .Big5 .big5
# 对于俄语，使用了多个字符集(如何使用主要依靠客户端):
AddCharset WINDOWS-1251 .cp-1251 .win-1251
AddCharset CP866 .cp866
AddCharset KOI8-r .koi8-r .koi8-ru
AddCharset KOI8-ru .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8 .utf8


# 下面的字符集没有映射到一个特定的标准(iso)上，但是它们在浏览器
# 中被广泛的支持。注意那些大写字母。
# (它不应该,但是它是为兼容一些浏览器而做)
#
# 参见 http://www.iana.org/assianments/character-sets 以取得
# 它们的列表。但是浏览器支持较少。
#
AddCharset GB2312 .gb2312 .gb
AddCharset utf-7 .utf7
AddCharset utf-8 .utf8
AddCharset big5 .big5 .b5
AddCharset EUC-TW .euc-tw
AddCharset EUC-JP .euc-jp
AddCharset EUC-KR .euc-kr
AddCharset shift_jis .sjis

#
# AddType 允许你为指定的文件类型添加或覆盖 mime.types 文件中配置的 MIME
#
AddType application/x-tar .tgz
AddType image/x-icon .ico

#
```

```
# AddHandler 允许你映射确定的文件扩展名到"handlers":
#  与文件类型无关的行为。这既能编译到服务器中也可以添加到 Action 指令中(看下面)。
#  为了在 ScriptAliased 指令指定目录的以外使用 CGI 脚本:
#(要使它可用，你还需要在 Options 中添加"ExecCGI"。)
#
# AddHandler cgi-script .cgi

#
#  对于那些包含他们自己的 HTTP 头的文件
#
# AddHandler send-as-is asis

#
#  对于 server-parsed imagemap 文件:
#
# AddHandler imap-file map

#
# agemap  文件:
#
#AddHandler imap-  文件映像

#
#  对于类型映像转移资源)
#(这是默认的设定以允许 Apache 的"It Worked"页能多种语言分发)。
#
AddHandler type-map var

#
#  过滤器允许你在将它发送到客户端前进行处理。
#
#  为了在服务器端分析包含(SSI)的.shtml 文档:
# (要执行这个指令，你还需要在 Options 指令中添加"Includes"。)
#
# AddType text/html .shtml
# AddOutputFilter INCLUDES .shtml

#
# Action 让你定义当调用匹配的媒体文件时将要执行的脚本。这将减少
#  那些经常使用的 CGI 脚本的 URL 路径名的重复输入。
#  格式:Action media/type /cgi-script/location
#  格式:Action handler-name /cgi-script/location
```

#

#
可配置的错误应答有三种风格:
1)plain text 2)local redirects 3) external redirects
#
一些示例:
ErrorDocument 500 "The server made a boo boo."
ErrorDocument 404 /missing.html
ErrorDocument 404 "/cgi-bin/missing_handler.pl"
ErrorDocument 402 http://www.panda.com/subscription_info.html
#

#
综合应用这些指令，我们可以创建一个国际化的出错应答。
#
我们使用 Alias 来重定向任意/error/HTTP_<error>.html.var 应答到
我们的多语言错误消息集合。使用正确的文本替代它。
#
通过加入下面的行，你就能够改变这些消息的显示，而不必改变
HTTP_<error>.html.var 文件。
#
Alias /error/include/ "/your/include/path/"
#
以将/usr/local/apache2/error/include/下的文件拷贝到/your/inclue/path/下
开始，你可以创建你自己的文件集合，甚至是其于每个虚拟主机的。
不管你的 ServerSignature 如何设置，默认的包含文件将显示你的
Aapche 版本号和你的 ServerAdmin 邮件地址
#
国际化的错误文档需要 mod_alias，mod_include 和 mod_negotiation 三个
模块。要激活它们，取消下面 30 行的注释符号

Alias /error/ "/usr/local/apache2/error/"
#
<Directory "/usr/local/apache2/error">
AllowOverride None
Options IncludesNoExec
AddOutputFilter Includes html
AddHandler type-map var
Order allow,deny
Allow from all
LanguagePriority en de es fr it nl sv

```
# ForceLanguagePriority Prefer Fallback
# </Directory>
#
# ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
# ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error.php
# ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
# ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
# ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
# ErrorDocument 410 /error/HTTP_GONE.html.var
# ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
# ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
# ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
# ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
# ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
# ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
# ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
# ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
# ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
# ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var

#
#  下面的命令更改标准的 HTTP 应答行为以处理已知的浏览器问题。
#
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

#
#  下面命令关闭对那些没有尾部"/"的目录的非 GET 请求的重定向，
#  这些命令修复了微软的采用 DAV 方法不能正确处理重定向的 WEB 文件夹的问题。
# Apple 下的 DAV 文件系统和 Gnome 下的 VFS 对 DAV 的支持也是采用这样的方法
#  进行处理的。
#
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully
BrowserMatch "^gnome-vfs" redirect-carefully

#
```

```
#  允许你使用 URL:http://servername/server-status 来通过 mod_status 生
#  成并报告服务器状态信息。改变.panda.com 为你自己的域名。
#
# <Location /server-status>
# SetHandler server-status
# Order deny,allow
# Deny from all
# Allow from .panda.com
# </Location>

#
#  允许使用 URL:http://servername/server-info 来远篁.??配置信息
# (需要 mod_info.c 支持)。改变".panda.com"为你自己的域名。
#
# <Location /server-info>
# SetHandler server-info
# Order deny,allow
# Deny from all
# Allow from .panda.com
# </Location>

#
#  代理服务器命令，去掉下面的行使代理服务可用。
#
# <IfModule mod_proxy.c>
# ProxyRequests On
# <Proxy *>
# Order deny,allow
# Deny from all
# Allow from .panda.com
# </Proxy>

#
#  安装或关闭 HTTP/1.1"通道"头处理。
# ("Full"添加服务器版本信息，"Block"移掉所有输出"通道"头信息。
#  可以设为下面各选项之一:Off | On | Full | Block
#
# ProxyVia On

#  最好为代理服务安装高速缓冲，去掉下面几行的注释符号:
# (没有 CacheRoot 则不缓冲)
#
```

CacheRoot "/usr/local/apache2/proxy"
CacheSize 5
CacheGcInterval 4
CacheMaxExpire 24
CacheLastModifiedFactor 01
CacheDefaultExpire 1
NoCache a-domain.com another-domain.edu joes.garage-sale.com

</IfModule>
代理命令结束。

#
附加的特定模块配置。
#
<IfModule mod_ssl.c>
Include conf/ssl.conf
</IfModule>

配置 MIME(multipropuse internet mail extend protocol)
AddHandler
SetHandler send-as-is # 按原状发送

CookieTracking CookieName CookieExpires
由 mod_usertrack.c 提供<非默认>
<IfModule mod_usertrack>
CookieTracking <on|off>
CookieName myCookie
</IfModule>
CookieExpires 600 #也可以用 CookieExpires "3 weeks 2 days 4 hours 22 seconds"
注意 CookieExpires 不可以在<Directory>中，该选项为全局

保护内容的安全性
User apache #运行的用户帐户
Group apache #运行的用户帐户组

<Directory>
<Location>
<Files>

使用 order,allow 和 deny 总是并行处理的
<Directory>
Order allow,deny

Deny from all
Allow 192.168.152.128/255.255.255.0
</Directory>

身份验证
AuthType <Basic|Digest> #验证方式
Basic 验证方式是用 BASE64 编码，即是明文传输的

AuthName "nico's zone" #验证区域名称
AuthUserFile /var/www/authenticaton/.htuser #用户密码文件,使用 htpasswd 创建
AuthGroupFile /var/www/authenticaton/.htuser #用户组文件,需要手工创建
require < user| group | valid-user > #需要满足的条件

mod_digest.c<非默认>
AuthDigestFile

使用数据库文件保持用户帐号和密码
使用 BSD DB2 格式，数据使用 crypt()加密
dbmmanage 可以创建 BSD DB2 和 DBM 格式的数据库

mod_auth_db.c <非默认>
AuthDBUserFile
AuthDBGroupFile

mod_auth_anon.c <非默认>
Anonymous #允许匿名登陆的用户名称
Anonymous_MustGiveEmail <on/off> #匿名用户必须提交 E-MAIL
Anonymous_LogEmail <on/off> #记录 E-mail 到 log 中
Anonymous_VerifyEmail <off/on> #检查 E-mail 的有效性
Anonymous_NoUserId <off/on> #允许不用用户名登陆

mod_digest.c<非默认>
不常用所以没有记录

Satisfy
对在封闭模块(如 <Directory>),中认证策略的满足条件
Satisfy <all|any>

<Limit> <LimitExcept>
<Limit> <LimitExcept>是逻辑上的对立
参数:CONNECT COPY DELETE GET(HEAD) OPTIONS LOCK MKCOL MOVE PATCH POST PROPFIND PROPPATCH PUT TRACE UNLOCK

第三区:虚拟主机
#
VirtualHost:你可以通过设置虚拟主机容器以实现在你的主机上保有多个
域名/主机名。大多数配置信息只使用基于名字的虚拟主机，因此服务器
不必担心 IP 地址的问题，下面的命令以*号代替虚拟主机名。
#
在你试着配置你的虚拟主机以前，请参见
URL:http://httpd.apache.org/docs-2.0/vhosts/>以取得更多的信息。
#
你可以使用命令行选项"-S"来检验你的虚拟主机配置。

#
使用基于名字的虚拟主机。
#
NameVirtualHost *

#
虚拟主机示例:
几乎所有的 Apache 命令都可以在虚拟主机容器中使用。
第一个虚拟主机区是用于向服务名未知的请求进行应答的配置。
#
<VirtualHost *>
ServerAdmin webmaster@dummy-host.panda.com
DocumentRoot /www/docs/dummy-host.panda.com
ServerName dummy-host.panda.com
ErrorLog logs/dummy-host.panda.com-error_log
CustomLog logs/dummy-host.panda.com-access_log commom
</virtualHost>
*****所有的虚拟主机

572 ☆虚拟目录

Alias /bbs "/var/www/bbs"
这个把 www.panda.com/bbs 定位到/var/www/bbs 而不是/var/www/html/bbs

重启 httpd

573 ☆主目录浏览

检查在文件/etc/httpd/conf/httpd.conf 中的 DocumentRoot 项目和下面的一样
DocumentRoot "/var/www/html"

目前/var/www/html 中没有任何主页文件

HTTP 浏览
开启 elinks 并且设定 URL 到:
http://panda.panda.com
如果您的浏览器在工作，您将看到缺省的服务器的索引页面。注意该文件并不是所存储的 HTML 文件，而是服务器在这些目录中没有缺省的 index.html 文件的时候自动生成的。

细心的用户可能会发现虽然在主目录设置了 Indexes 权限，且主目录中并不存在默认文档，但访问时并不会出现目录列表，而只出现 Apache 的测试页面。解决这个问题的方法很简单，只要将位于/etc/httpd/conf.d/目录下的 welcome.conf 文件删除重启 Apache 即可。这时候就会出现 index
这个选择很危险

574 ☆Apache 虚拟主机

用 Apache 设置虚拟主机服务通常可以采用两种方案:基于 IP 地址的虚拟主机和基于名字的虚拟主机

httpd -S
检查虚拟主机的配置

基于 IP 的虚拟主机
设置实现基于 IP 地址的虚拟主机服务实现前提:
这种方式需要在机器上设置 IP 别名，也就是在一台机器的网卡上绑定多个 IP 地址去为多个虚拟主机服务。而且要使用这项功能还要确定在你的 LINUX 内核中必须支持 IP 别名的设置，否则你还必须重新编译内核。

配置步骤
www.panda.com 192.168.152.128
让 ISP 作好相应的域名解析工作
为网卡设置 IP 别名
/sbin/ifconfig eth0:0 192.168.152.128 netmask 255.255.255.0

设置"/etc/httpd/conf/httpd.conf",在文件中加入:
<VirtualHost 192.168.152.128>
ServerName panda.panda.com
ServerAdmin webmaster@panda.com
DocumentRoot /var/www/virtual/panda.panda.com　　#为 www.panda.com
ServerAlias panda.com
ErrorLog /var/log/httpd/panda.panda.com/error.log
CustomLog /var/log/httpd/panda.panda.com/access_log combined
<Directory /var/www/virtual/panda.panda.com/html> #panda.panda.com/html 的设置

Options Indexes Includes
</Directory>
</VirtualHost>

建立相应的目录
mkdir /var/log/httpd/panda.panda.com/
mkdir -p /var/www/virtual/panda.panda.com/html
cd /home/httpd/panda.panda.com/html

cat > index.html <<EOF
panda.panda.com
EOF

这时不要忘记设置权限，(包括文件的)切记！！
chmod -R 755 /var/www/virtual/panda.panda.com/html
只要其他用户的权限为 5 就是 r-x 的权限

不利因素
这种虚拟主机的实现方法有一个严重的不足，那就是，每增加一个虚拟主机，就必须增加一个 IP 地址。而由于 IP 地址空间已经十分紧张，所以通常情况下是无法取得这么多的 IP 地址的。而且从某种意义上说，这也是一种 IP 地址浪费。

基于域名的虚拟主机
修改配置文件"/etc/httpd/conf/httpd.conf"，在这个配置文件中增加以下内容
NameVirtualHost 192.168.152.128
<VirtualHost 192.168.152.128>
ServerName panda.panda.com
ServerAdmin webmaster@panda.com
DocumentRoot /var/www/virtual/www1.panda.com
ServerAlias panda.com
ErrorLog /var/log/httpd/www1.panda.com/error.log
</VirtualHost>

<VirtualHost 192.168.152.128>
ServerName www1.panda.com #不同的虚拟主机名
ServerAdmin webmaster@panda.com
DocumentRoot /var/www/virtual/www1.panda.com
ErrorLog /var/log/httpd/www1.panda.com/error.log
</VirtualHost>

也就是在基于 IP 地址的配置基础上增加一句:NameVirtualHost 192.168.152.128 而已。在本例中，为了体现只需要增加一次，所以特别地设置了两个虚拟主机服务。

最后也是建立相应的目录，将主页内容放到相应的目录中去就可以了
要在 DNS 中加记录

VirtualDocumentRoot
mod_vhost_alias.c 模块提供
大大简化了配置虚拟主机

一个好的管理员会为没个虚拟站点分别设置不同的的安全性设置

575 ☆Apache 的 CGI 环境变量

服务器变量
服务器变量由 Apache 设置用来通知 CGI 程序有关 Apache 的情况。通过使用这些变量，CGI 程序能确定有关服务器的不同信息:Apache 的版本，管理员的 E-Mail 地址等。

SERVER_SOFTWARE
这个变量是 WWW 服务器 Apache 的版本号，它的值形如:Apache/Version，如 Apache/1.3;

GATEWAY_INTERFACE
这个变量的值是当前 CGI 规范的版本号，其值形如:CGI/1.1;

SERVER_ADMIN
如果在 httpd.conf 文件中有设置站点管理员的 e-mail 地址的话，这个变量就会存放着这个 e-mail 地址;

DOCUMENT_ROOT
这个变量存放在是被访问的 WWW 站点的 DocumentRoot 命令指定的值。

客户请求变量
Apache 提供的有关客户请求方的环境变量有许多，以下只是有选择性地介绍一些最常见的。

SERVER_NAME
此变量可以告诉 CGI 程序它访问的是哪一个主机。这个值可以是 IP 地址也可以是完整的主机名;

HTTP_ACCEPT
此变量被赋值为客户所能接受的 MIME 类型的列表，如:HTTP_ACCEPT=image/gif;

HTTP_ACCEPT_CHARSET
此变量被赋值为客户所能接受的字符集，如:
HTTP_ACCEPT_CHARASET=iso-8859-1.,*,utf-8;

HTTP_ACCEPT_LANGUAGE

此变量被赋值为客户所能接受的语言，如:HTTP_ACCEPT_LANGUAGE=en;

HTTP_ACCEPT_AGENT
这个变量指定发出请求的系统正在运行的浏览器类型和操作系统;

HTTP_PORT
服务端口;

REMOTE_HOST
客户端的 IP 地址或 IP 名称信息;

REMOTE_PORT
客户端的端口号;

576 ☆Apache 的 CGI 配置

rpm -q perl

httpd.conf 中
把这一行的注释去掉
#AddHandler cgi-script .cgi

并且目录下加
Options ExecCGI

这样可以在 ScriptAlias 以外的目录中执行脚本

如果只限定在某个目录下执行脚本
在<VirtualHost> 块中增加一行:
ScriptAlias /cgi-bin/ /var/www/virtual/panda.panda.com/cgi-bin/

建立 test.sh
#!/bin/bash
echo Content-Type: text/html;
echo
echo "<pre>"
echo `cat /etc/passwd`
echo "</pre>"

建立 test.cgi
#!/usr/bin/perl
print "Content-type: text/html\n\n"; 这一句很重要

print "hello";

使得该脚本对于用户，组和其他可读并且可执行:
chmod 555 test.sh

或者,然后
chmod +x test.cgi

Apache 服务器为用户提供的两种 cgi-bin 访问方法。
1)使用 Directory 或 DirectoryMatch 容器
当在配置文件 httpd.conf 中用 UserDir 命令被赋值为目录名称时，Apache 就把它作为用户 WWW 站点的顶层目录。
例如:
UserDir Public_html
当 Apache 接到 www.panda.com/~user 的请求，就到/home/user/Public_html 取出主页发送给客户。

如果要为每个用户添加 CGI 支持就在 Apache 的配置文件 access.conf 中添加下列配置:
<DirectoryMatch "/home/[a-z]+/public_html/cgi-bin">
Options ExecCGI
AddHandler cgi-script .cgi .pl
</DirectoryMatch>
注:将 DirectoryMatch 换成 Directory 亦可
在这种方法中，Apache 服务器将 www.panda.com/~user/cgi-bin 请求翻译成为了 /home/user/Public_html/cgi-bin/，并允许执行任何带有正确扩展名(.cgi 或.pl)的 CGI 程序。

2)使用 ScriptAliasMatch 命令
通过使用 ScriptAliasMatch 命令，也可以为每个用户添加 CGI 支持。例如:
ScriptAliasMatch ~([a-z]+)/cgi-bin/(.*) /home/$1/public_html/cgi-bin/$2
这个命令将用户名与$1 相匹配，其中$1 与~([a-z]+)相等。将/cgi-bin/后面的任何内容与$2 相匹配，其中$2 与(.*)相等。
这个设置也就实现了将 www.panda.com/~user/cgi-bin/test.cgi 请求解释为:
/home/user/Public_html/cgi-bin/test.cgi

ScriptAliasMatch ~([a-z]+)/cgi-bin/(.*) /home/httpd/public/cgi/$2
应该是:/home/httpd/public/cgi/test.cgi

577 ☆Apache 的 PHP 配置

rpm -q php

在 Apache 主配置文件 httpd.conf 中默认有一条"Include conf.d/*.conf"语句,它的含义是将目录 /etc/httpd/conf.d/的所有*.conf 文件包含到 httpd.conf 中.PHP 解释器的安装程序会自动在目录 /etc/httpd/conf.d/建立一个名为 php.conf 的配置文件,这个文件包含了 PHP 的配置选项.
由于历史原因,许多原来基于 PHP3 的程序文件扩展名为.php3,为了能让这些 PHP3 的程序文件运行,应该在 php.conf 文件中为.php3 扩展名的文件建立映射.编辑/etc/httpd/conf.d/php.conf,找到语句 "AddType application/x-httpd-php .php",将其改为“AddType application/x-httpd-php .php .php3"
不需要 ExecCGI 权限

test.php
<? phpinfo(); ?>

578 ☆apachectl

#要加密:apachectl startssl
#不加密:apachectl start

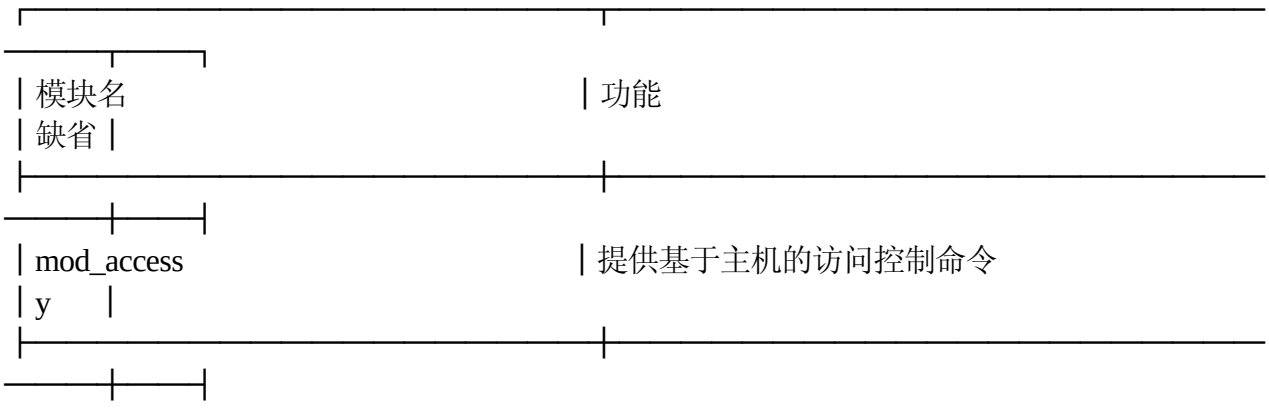
测试配置是否正确
apachectl configtest
同 httpd -t

启动 apache
apachectl start 或者 httpd -k start

重起 apache
apachectl graceful 或者 httpd -k graceful

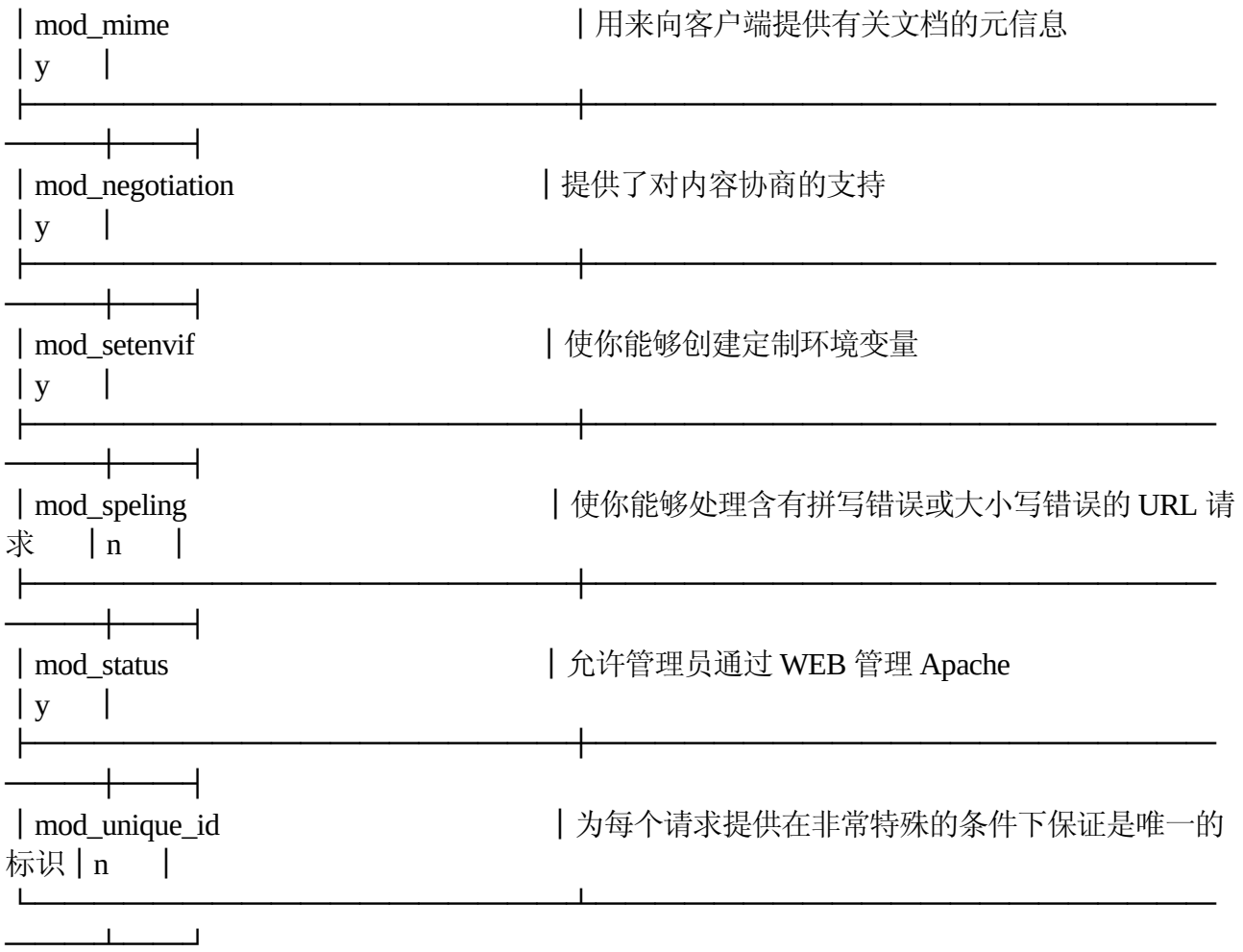
停止 apache
apachectl stop 或者 httpd -k stop

579 ☆Apache 的模块功能



mod_actions	能够运行基于 MIME 类型的 CGI 脚本或 HTTP 请求方法	y
mod_alias	能执行 URL 重定向服务	y
mod_asis	使文档能在没有 HTTP 头标的情况下被发送到客户端	y
mod_auth	支持使用存储在文本文件中的用户名、口令实现认证	y
mod_auth_dbm	支持使用 DBM 文件存储基本 HTTP 认证	n
mod_auth_mysql	支持使用 MySQL 数据库实现基本 HTTP 认证	n
mod_auth_anon	允许以匿名方式访问需要认证的区域	y
mod_auth_external	支持使用第三方认证	n
mod_autoindex	当缺少索引文件时，自动生成动态目录列表	y
mod_cern_meta	提供对元信息的支持	n
mod_cgi	支持 CGI	y

mod_dir	能够重定向任何对不包括尾部斜杠字符命令的请求
y	
mod_env	使你能够将环境变量传递给 CGI 或 SSI 脚本
n	
mod_expires	让你确定 Apache 在服务器响应请求时如何处理 Expires
y	
mod_headers	能够操作 HTTP 应答头标
y	
mod_imap	提供图形映射支持
n	
mod_include	使支持 SSI
n	
mod_info	对服务器配置提供了全面的描述
y	
mod_log_agent	允许在单独的日志文件中存储用户代理的信息
n	
mod_log_config	支持记录日志
y	
mod_log_referer	提供了将请求中的 Referer 头标写入日志的功能
n	



580 ☆Apache 的代理服务器

Apache 作为 WWW 服务器软件，在内部提供了 HTFP 代理功能。

1.前向代理服务器
前向代理服务器通常位于用户主机和要访问的远程网络之间。它从远程服务器取得所要求的资源，然后返回给用户，同时存在磁盘上，以供下次使用。
在这种情况下，客户端的主机知道它们正在使用代理服务器，因为每个主机都必须配置为使用代理服务器。
这类代理服务器也称为缓冲代理服务器。逆向服务器也可以缓冲数据，但它的作用恰好与前向服务器相反。

2.逆向代理服务器
逆向代理服务器位于互联网资源前面，逆向服务器从原始服务器找到被请求的资源，并反它返回给用户主机。
与前向代理服务器不同的是，逆向代理服务器的用户并不知道它们连接的是代理服务器而不是资源服务器本身。

581 ☆Apache 配置代理服务器

为了允许 Apache 作为代理服务器，需要将 ProxyRequests 设为 On，然后根据你希望代理服务器做什么而增加什么附加配置。无论你希望做什么，你所选的代理配置都应该放入一个特殊<Directory>容器中。

<Directory proxy:*>
</Directory>

修改文件/etc/httpd/conf/httpd.conf，在其中添加与代理和缓存相关的功能。

ProxyRequests On/Off

启用或者禁用 Apache 代理功能

Proxyremote path URL remote server

定义此代理服务器的远程代理。当用户请求与 URL 匹配时，就使用 remote server 作为远程代理服务器。

其中 remote server 的格式是: protocol://hostname[:port]，由于 Apache 只能代理 HTTP 服务，所以 protocol 值恒为 HTTP

ProxyPass path URL

允许把远程服务器镜像到本地服务器中。这时，本地代理服务器好像是远程代理服务器的一个镜像

ProxyBlock word/hostname/domain

代理服务器过滤功能。在 ProxyBlock 关键字以后定义了一组词语、节点名称和域名。如果用户的 HTYP 请求中包含了这里的词语、节点名称或者域名，请求将被过滤掉

CacheRoot directory

代理缓存的根目录

CacheSize size

代理缓存大小，以 KB 为单位

CacheGcInterval time

每隔 time 小时检查缓存区，如果缓存占用空间超过 CacheSize 设置的上限，就删除文件缓存中的文件最多保存 time 小时，这里定义了文件的过期时间

CacheLastModifiedFactor factor

如果没有定义文件过期时间，就按照下面的公式计算:过期时间=最近一次修改的时间间隔*factor

CacheDirLevels levels

缓存中子目录的层数

CacheDirLenSth lenSth

代理缓存子目录名的字母数

CacheDefauhExpire time

如果文件是通过一个不支持过期时间的协议获取的，则使用 time 作为过期时间

NoCache word/hostname/domain

在 NoCache 关键字以后定义了一组词语、节点名称和域名。包含这些词语、节点名称或者域名的 HTFP 文件将不被缓存

修改文件/etc/httpd/conf/httpd.conf，在其中添加对代理目录的访问控制。下面是一个实例。
<Directory proxy:*>
<Limit GET PUT DELECT CONNECT POST OPTIONS>
order deny,allow
deny from all
allow from .panda.com
</Limit>
</Directory>

确认 CacheRoot 关键字指定的缓存目录已经存在。
重新启动 httpd，缓存功能就可以生效了。

582 ☆Apache 配置逆向代理服务器

把 WWW 服务器设在网络内部的方法可以把 WWW 服务器与外部世界隔离，提高安全性。这时，我们需要在防火墙上也安装 Apache 服务器，使用它提供对 WWW 服务器的代理访问。同时，结合我们前面介绍的虚拟主机技术，把防火墙作为 Apache 虚拟主机，使得防火墙上的主页不会被访问到。

内部局域网中的 WWW 服务器 IP 地址为 192.169.152.128，防火墙主机内部 IP 地址为 192.169.152.129，外部 IP 地址为 202.100.0.1。

配置防火墙上 Apache 的步骤如下。

(1)在/etc/httpd/conf/httpd.conf 文件中添加虚拟主机配置。
NameVirtualHost 202.100.0.1
servername www.panda.com
errorlog/var/log/httpd/error-log
transferlog/var/log/httpd/access-log
rewriteengine On
proxyrequests off
usecanonicalname off
rewriterule^/(.*)$http://192.169.152.128$ 1 [P,L]

(2)配置局域网中的 DNS 服务器，把 www.panda.com 指向 202.100.0.1。这样，所有对 IP 地址 202.100.0.1 的访问都会被重新导向到内部 WWW 服务器中。我们还需要对局域网内部 WWW 服务器的配置进行修改，具体步骤如下。

1]Apache 采用默认配置，主目录为/home/httpd/html，主机域名为 panda.com.cn，别名为 www.panda.com，并且在 srm.conf 中添加以下别名
定义。
Alias /pub/home /ftp/pub/

DefaultType application/octet-stream

2]在/etc/httpd/conf/httpd.conf 中增加一项。
Options lndexes
AllowOverride AuthConfiq
order allow,deny
allow from all

3]在/home/ftp/pub 目录下放人.htaccess 文件，内容如下。
#cat.htaccess
AuthName Branch Office Public Software Download Area
AuthType Basic
AuthUserFile /etc/.usrpasswd
require valid-user

4]执行如下命令:
＃htpasswd -c /etc/.usrpasswd user1

分别创建允许访问/pub 文件服务的不同的外部用户名和口令。

583 ☆Apache 配置缓冲服务器

ProxyRequest 设置为 On，并且不需要附加其他配置，所有请求均可由这台代理服务器代理服务。

让 Apache 允当远程 WWW 站点的缓冲
第一步:将 ProxyRequest 设置为 On
第二步:创建配置如下:
<Directory proxy:*>
CacheRoot /www/cache
CacheSize 1024
CacheMaxExpire 24
</Directory>
这里的意思是设置 Cache 目录为/www/cache;大小为 1024KB，即 1MB;缓冲中的内容在 24 小时后失效。

584 ☆Apache 镜像站点

建立镜像站点(其实这也就是所谓的逆向代理服务器)
将 ProxyRequest 设置为 On
创建配置如下:
<Directory proxy:*>
ProxyPass /www.panda.com /

CacheRoot /www/cache
CacheDefaultExpire 24
</Directory>

用 Apache 作代理服务器的性能并不高，效果并不好。不建议使用。

ProxyRequests On

<Proxy *>
Order deny,allow
Allow from all
</Proxy>

CacheEnable disk /
CacheRoot "/var/cache/mod_proxy"

585 ☆htpasswd

密码文件创建程序

建立用户文件
htpasswd -c /etc/httpd/.htpasswd 用户名
-c 表示删去原来的文件,重建,后面增加用户就不用这个了
如果你们想修改密码，可以如下
htpasswd -m .htpasswd panda

注意:/etc/httpd/必须是网络不能访问到的路径,以免被下载

建立组文件

/etc/httpd/.htgroup 为用户组的文件名，文件内容格式为
admin:user user1
其中的用户要先用 htpasswd 建立

设置文件权限，确保 Apache 用户有读的权限(不必要)
chgrp apache .htpasswd
chmod g+r .htpasswd

586 ☆Apache 的安全机制

基于主机的认证方式

在种认证模式顾名思义，访问是用主机名或主机 IP 地址来控制的。支持这种认证方式的是 Apache 的 mod_access 模块，这个模块缺省状态下是被安装了的。该模块用以下几种 Apache 命令来提供访问控制功能。

order 命令
语法:order deny,allow | allow,deny
这个命令定义评价 allow 和 deny 命令的先后顺序。
Order 选项用于定义缺省的访问权限与 Allow 和 Deny 语句的处理顺序。Allow 和 Deny 语句可以针对客户机的域名或 IP 地址进行设置，以决定哪些客户机能够访问服务器。Order 语句通常设置为以下两种值之一。
• allow,deny：缺省禁止所有客户机的访问，且 Allow 语句在 Deny 语句之前被匹配。如果某条件既匹配 Deny 语句又匹配 Allow 语句，则 Deny 语句会起作用(因为：Deny 语句覆盖了 Allow 语句)。
• deny,allow：缺省允许所有客户机的访问，且 Deny 语句在 Allow 语句之前被匹配。如果某条件既匹配 Deny 语句又匹配 Allow 语句，则 Allow 语句会起作用(因为 Allow 语句覆盖了 Deny 语句)。

allow 命令
语法:allow from host1 host2 host3 ...
这个命令定义了允许访问站点或目录的主机清单。主机清单可以用以下几种形式表示:
ALL:代表所有主机;
主机的全域名，如:www.panda.com;
主机的部分域名，如:.mot.com;
完整的 IP 地址，如:202.98.2.32;
部分 IP 地址，如:202.98
网络地址/网络掩码对，如:202.98.0.0/255.255.0.0
网络地址/nn(CIDR 定义)，如:202.98.0.1/16

deny 命令
语法:deny from host1 host2 host3 ...
这个命令定义了禁止访问站点或目录的主机清单，其他与 allow 命令相似。

基于用户名/口令的认证方式
这种认证方式其实相当简单，当 WWW 浏览器请求经此认证模式保护的 URL 时，将会出现一个对话框，要求用户键入用户名和口令。用户输入后，传给 WWW 服务器，WWW 服务器验证它的正确性，如果正确，返回页面，否则返回 401 错误。要说明的一点是，这种认证模式是基本的，并不能用于安全性要求极高的场合。

Apache 中有许多模块可以支持这种认证方式，下面我们就介绍一下最基本、最标准的 mod_auth 模块。正如前面提到的一样，mod_auth 模块使用存储在文本文件中的用户名、组名和口令来实现认证。这种方法非常适合处理少量用户，它能工作得很好。如果你需要对大量的用户，如数以千计的用户做认证时，这种方法的性能将急剧下降到不可忍受，所以当这种情况下，就需要考虑使用 mod_dbm 模块或 mod_mysql 模块来获得更好的性能。

修改/etc/httpd/conf/httpd.conf 文件
基本情况: www.panda.com 的站点有设置为:
DocumentRoot /var/www/virtual/panda.panda.com/
AccessFileName .htaccess
AllowOverride All

Alias /se "/var/www/virtual/panda.panda.com/se"
<Directory "/var/www/virtual/panda.panda.com/se">
Options Indexes MultiViews
AllowOverride AuthConfig #表示进行身份验证 这是关键的设置
Order deny,allow
Allow from all
</Directory>

并在虚拟目录 se 中添加.htaccess 文件
AuthType 选项定义了对用户实施认证的类型，最常用的是由 mod_auth 提供的 Basic。
AuthName 选项定义了 Web 浏览器显示输入用户/密码对话框时的领域内容。
AuthUserFile 选项定义了口令文件的路径，即使用 htpasswd 建立的口令文件。
Require user 选项定义了允许哪些用户访问，各用户之间用空格分开。

用 vi 在/var/www/virtual/panda.panda.com/se 目录下建立一个文件.htaccess，写入以下几行:
AuthName "My Friend Only" (注:这个名字是任取的)
AuthType Basic
AuthUserFile /etc/httpd/.htpasswd #htpasswd 生成的文件
或者 AuthGroupFile /home/httpd/.htgruop
Require valid-user #限制是所有合法用户还是指定用户或组
或者 Require user panda
或者 Require group admin
#密码文件推荐使用.htpasswd,因为 apache 默认系统对".ht"开头的文件默认不允许外部读取，安全系数会高一点。

为了服务器的性能，一般不推荐使用 AllowOverride AuthConfig 或者 AllowOverride ALL，因为这会使服务器会不断的去寻找.htaccess,从而影响服务器的效能，一般我们把一些后台管理界面或者其他特殊目录可能需要加验证这个需求。

AllowOverride 选项用于定义位于每个目录下的.htaccess(访问控制)文件中的指令类型。基于安全和效率的原因，虽然可以通过.htaccess 来设置目录的访问权限，但应尽可能地避免使用.htaccess 文件，所以一般将 AllowOverride 设置为“None”，禁止使用.htaccess 文件，而将目录权限的设置放在主配置文件 httpd.conf 的<Directory>和</Directory>语句之间。

587 ☆Apache 目录授权

apache 默认的用户和组为为 apache
将站点的目录和文件的权限赋予 apache
find /var/www -exec chown apache {} \;
find /var/www chgrp apache {} \;
find /var/www -type d -exec chmod 755 {} \;
find /var/www -type f -exec chmod 640 {} \;
给适当的文件适当的权限是一个管理员的责任所在。

588 ☆Apache 的用户个人主页

首先，为需要个人主页空间的员工在 LINUX 上开设一个帐号。这样，它就拥有了一个用户主目录"/home/用户帐号名"。
adduser 用户帐号名
passwd 用户帐号名

在用户主目录下建立一个目录"public_html"，然后为其设置相应的权限。
cd ~用户帐号名
mkdir /home/用户名/public_html
chmod 755 -R /home/用户名

确认在 httpd.conf 文件中的 UserDir 命令设置的是 public_html 目录。让员工将自己的个人主页上传到自己用户主目录下的 public_html 目录中。
UserDir=public_html

现在就可以使用"http://www.panda.com/~用户帐号名"来访问员工的个人主页了。

589 ☆HTTPS

/etc/httpd/conf.d/ssl.conf
中的
Listen 443
https 用 TCP/UDP 443 端口

httpd.conf 是 apache 的主配置文件。apache 的服务相关的东西全在这里面配
而 ssl.conf 是 ssl 的配置文件，这里配置密钥和密钥相关的东西

你可以用 httpd -l 来查看一下是否有 ssl
httpd -tS
检查虚拟主机配置

另外还要在 httpd.conf 里指定 ssl_mod 的路径
LoadModule ssl_module <路径>/ssl_mod.so

还要在 httpd.conf 里指定 ssl.conf 的路径
Include <路径>/ssl.conf

改以下 4 行：
DocumentRoot "/www/docs/www.panda.com"
ServerName www.panda.com:443
SSLCertificateKeyFile /etc/httpd/conf/server.crt
SSLCertificateKeyFile /etc/httpd/conf/server.key

之后，在/etc/httpd/conf 目录下运行 make server.crt server.key

以下 2 行：
DocumentRoot "/www/docs/www.panda.com"
ServerName www.panda.com:443
在/etc/httpd/conf/httpd.conf 中有对应项。
要加入 NameVirtualHost *:443

apachectl startssl
启动

590 ☆Apache 查看状态

报告服务器的状态可以用命令 apachectl status

server-status
由模块 mod_status.c 提供
但是 server-status 可以提供更多的信息
http://servername/server-status 来访问
ExtendedStatus On
<Location /server-status>
SetHandLer server-status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>

server-info
由模块 mod_info.c 提供
http://servername/server-info 来访问

<Location /server-info>
SetHandLer server-info
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>

591 ☆Squid

592 ☆Squid 启动停止

service squid start

配置您的浏览器使用您的 localhost 作为您的 Proxy 并且把端口设定为 3128。

593 ☆Squid 防火墙

iptables -I INPUT -p tcp --dport 3128 -j ACCEPT

594 ☆Squid 配置文件

- /etc/squid/ 配置目录
- /var/log/squid/ 日志
- /var/spool/squid/ 缓冲目录

595 ☆Squid 配置

/etc/squid/squid.conf
注意:这个文件中的参数和选项是区分大小写的

内部网络接口 eth0 的 IP 地址为 192.168.152.128，外部网络接口 eth1 的 IP 地址为 202.100.0.1。下面是一个基本的代理所需要配置选项:

http_port 192.168.152.128:3128
默认监听端口是 3128，当然也可以是任何其它端口，只要不与其它服务发生冲突即可。为了安全起见，在前面加上 IP 地址，Squid 就不会监听外部的网络接口。

cache_mgr root@panda.com
是服务器管理者的电子邮件，当错误发生时，该地址会显示在错误页面上，便于用户联系:

以下这些参数告诉 Squid 缓存的文件系统、位置和缓存策略:

cache_dir ufs /var/squid
硬盘缓冲大小,cache_dir ufs /var/squid 4096 16 256,ufs 为缓冲存储类型,大小为 4096MB,第一级子目录最多 16 个,第二级子目录最多 256 个

cache_mem 32MB
做缓冲的物理内存,不要超过总内存的 1/3
cache_swap_low 90
cache_swap_high 95

在这里，Squid 会将/var/squid 目录作为保存缓存数据的目录，每次处理的缓存大小是 32 兆字节，当缓存空间使用达到 95%时，新的内容将取代旧的而不直接添加到目录中，直到空间又下降到 90%才停止这一活动。如果不想 Squid 缓存任何文件，如某些存储空间有限的专有系统，可以使用 null 文件系统（这样不需要那些缓存策略）:
cache_dir null /tmp

cache_effective_user squid
cache_effective_group squid
squid 代理使用的用户和组,如果没有此用户,要建立用户

dns_nameservers 10.0.0.1 192.172.0.4
可以让 squid 解析服务名

下面的几个关于缓存的策略配置中，较主要的是第一行，即用户的访问记录，可以通过分析它来了解所有用户访问的详尽地址:

cache_access_log /var/squid/access.log
用户的访问 internet 的详细信息,可以看每台客户机上网记录
cache_log /var/squid/cache.log
记录缓存相关信息
cache_store_log /var/squid/store.log
记录网页在缓存中的调用情况

下面这行配置是在较新版本中出现的参数，告诉 Squid 在错误页面中显示的服务器名称:
visible_hostname No1.proxy

以下配置告诉 Squid 如何处理用户，对每个请求的 IP 地址作为单独地址处理:
client_mask 255.255.255.255

596 ☆Squid 配置透明代理

但是很多 Squid 都被用来做透明代理。所谓透明代理，就是客户端不知道有代理服务器的存在，当然也不需要进行任何与代理有关的设置，从而大大方便了系统管理员。相关的选项有以下几个:

/etc/squid/squid.conf 中需要有:
httpd_accel_host virtual
加速模式,虚拟主机模式
httpd_accel_port 80
要加速的请求端口
httpd_accel_with_proxy on
即是 web 请求加速器又是缓存代理服务器
httpd_accel_user_host_header on
存储对象加上主机名而不是 ip 地址

在 Linux 上，可以用 iptables/ipchains 直接将对 Web 端口 80 的请求直接转发到 Squid 端口 3128，由 Squid 接手，而用户浏览器仍然认为它访问的是对方的 80 端口。例如以下这条命令:
iptables -t nat -A PREROUTING -s 192.168.0.200/32 -p tcp --dport 80 -j REDIRECT 3128
就是将 192.168.0.200 的所有针对 80 端口的访问重定向到 3128 端口。

597 ☆Squid 访问控制列表

ACL 的基本格式如下:
acl 列表名称 列表类型 [-i] 列表值

列表名称
区分各个 squid 列表,不可同名

列表类型
src
源 ip,客户
dst
目标 ip,服务器
srcdomain
源域名称
dstdomain
目标域名称
time
一天中的时刻和一周中的一天,语法为:[星期][时间段]
[星期]M|T|W|H|F|A|S
[时间段]10:00-20:00
url_regex
URL 规则表达式
urlpath_regex:URL-path
省略去主机名和协议的 URL 规则表达式
proxy_auth

外部程序用户验证
maxconn
单一 ip 最大连接数

-i
忽略列表值的大小写,否则区分大小写

列表值
不同列表类型的列表值不同

允许或拒绝某个列表的 http 请求
http_access [allow|deny] 访问控制列表名称
是顺序读取的
如果拒绝在前面就先拒绝,允许在前面就允许.
http_access allow .panda.com
http_access deny www.panda.com
这个会接收.panda.com 域下的所有,调换次序后,会先拒绝 www 主机

所有设置完成后，关键且重要的任务是访问控制。Squid 支持的管理方式很多，使用起来也非常简单（这也是有人宁愿使用不做任何缓存的 Squid，也不愿意单独使用 iptables 的原因）。Squid 可以通过 IP 地址、主机名、MAC 地址、用户/密码认证等识别用户，也可以通过域名、域后缀、文件类型、IP 地址、端口、URL 匹配等控制用户的访问，还可以使用时间区间对用户进行管理，所以访问控制是 Squid 配置中的重点。Squid 用 ACL（Access Control List，访问控制列表）对访问类型进行划分，用 http_access deny 或 allow 进行控制。根据需求首先定义两组用户 advance 和 normal，还有代表所有未指明的用户组 all 及不允许上网的 baduser，配置代码如下:

acl advance 192.168.152.100-192.168.152.128/32
acl normal src 192.168.152.128-192.168.152.200/32
acl baduser src 192.168.152.128/32
acl baddst dst www.panda.com
acl all src 0.0.0.0/0

http_access deny baduser
http_access allow advance
http_access allow normal

访问控制列表中 squid 默认拒绝所有客户机访问.

比如 acl all src 0.0.0.0/0，其名称是 all，控制方式是 src 源 IP 地址，控制目标是 0.0.0.0/0 的 IP 地址，即所有未定义的用户。出于安全考虑，总是在最后禁止这个列表。

下面这个列表代表高级用户，包括 IP 地址从 192.168.0.2 到 192.168.152.128 的所有计算机:

acl advance 192.168.0.2-192.168.152.128/32

下面这个 baduser 列表只包含一台计算机，其 IP 地址是 192.168.152.128:
acl baduser 192.168.152.128/32

ACL 写完后，接下来要对它们分别进行管理，代码如下:
http_access deny baduser
http_access allow advance
http_access allow normal

上面几行代码告诉 Squid 不允许 baduser 组访问 Internet，但 advance、normal 组允许（此时还没有指定详细的权限）。由于 Squid 是按照顺序读取规则，会首先禁止 baduser，然后允许 normal。如果将两条规则顺序颠倒，由于 baduser 在 normal 范围中，Squid 先允许了所有的 normal，那么再禁止 baduser 就不会起作用。

特别要注意的是，Squid 将使用 allow-deny-allow-deny……这样的顺序套用规则。例如，当一个用户访问代理服务器时，Squid 会顺序测试 Squid 中定义的所有规则列表，当所有规则都不匹配时，Squid 会使用与最后一条相反的规则。就像上面这个例子，假设有一个用户的 IP 地址是 192.168.0.201，他试图通过这台代理服务器访问 Internet，会发生什么情况呢？我们会发现，他能够正常访问，因为 Squid 找遍所有访问列表也没有和 192.168.0.201 有关的定义，便开始应用规则，而最后一条是 deny，那么 Squid 默认的下一条处理规则是 allow，所以 192.168.0.201 反而能够访问 Internet 了，这显然不是我们希望的。所以在所有 Squid.conf 中，最后一条规则永远是 http_access deny all，而 all 就是前面定义的"src 0.0.0.0"。

假如不想让用户访问某个网站应该怎么做呢？可以分为两种情况:一种是不允许访问某个站点的某个主机，比如 ok 的主机是 ok.sina.com.cn，而其它的新浪资源却是允许访问的，那么 ACL 可以这样写:
acl sinapage dstdomain ok.sina.com.cn
... ...
http_access deny ok
... ...
由此可以看到，除了 ok，其它如 http://www.sina.com.cn、news.sina.com.cn 都可以正常访问。

另一种情况是整个网站都不许访问，那么只需要写出这个网站共有的域名即可，配置如下:
acl qq dstdomain .panda.com.cn
注意 tcccent 前面的".",正是它指出以此域名结尾的所有主机都不可访问，否则就只有 panda.com.cn 这一台主机不能访问。

如果想禁止对某个 IP 地址的访问，如 202.118.2.182，可以用 dst 来控制，代码如下:
acl badaddr dst 202.118.2.182
当然，这个 dst 也可以是域名，由 Squid 查询 DNS 服务器将其转换为 IP。

还有一种比较广泛的控制是文件类型。如果不希望普通用户通过代理服务器下载 MP3、AVI 等文件，完全可以对他们进行限制，代码如下:
acl mmxfile urlpath_regex \.mp3$ \.avi$ \.exe$
http_access deny mmxfile

看到 regex，很多读者应该心领神会，因为这条语句使用了标准的规则表达式（又叫正则表达式）。它将匹配所有以.mp3、.avi 等结尾的 URL 请求，还可以用-i 参数忽略大小写，例如以下代码:
acl mmxfile urlpath_regex -i \.mp3$
这样，无论是.mp3 还是.MP3 都会被拒绝。当然，-i 参数适用于任何可能需要区分大小写的地方，如前面的域名控制。

如果想让普通用户只在上班时间可以上网，而且是每周的工作日，用 Squid 应当如何处理呢？看看下面的 ACL 定义:
acl worktime time MTWHF 8:30-12:00 14:00-18:00
http_access deny !worktime
首先定义允许上网的时间是每周工作日（星期一至星期五）的上午和下午的固定时段，然后用 http_access 定义所有不在这个时间段内的请求都是不允许的。

或者为了保证高级用户的带宽，希望每个用户的并发连接不能太多，以免影响他人，也可以通过 Squid 控制，代码如下:
acl conncount maxconn 3
http_access deny conncount normal
http_access allow normal
这样，普通用户在某个固定时刻只能同时发起三个连接，从第四个开始，连接将被拒绝。

更多的控制方式可以参考 squid.conf.default
把特殊的 ACL 的 http_access 放在前面

598 ☆Squid 访问控制列表使用 MAC

通过 IP 地址来识别用户很不可靠，比 IP 地址更好的是网卡的 MAC 物理地址。要在 Squid 中使用 MAC 地址识别，必须在编译时加上"--enable-arp-acl"选项，然后可以通过以下的语句来识别用户:
acl advance arp 00:01:02:1f:2c:3e 00:01:02:3c:1a:8b ...

它直接使用用户的 MAC 地址，而 MAC 地址一般是不易修改的，即使有普通用户将自己的 IP 地址改为高级用户也无法通过，所以这种方式比 IP 地址可靠得多。

599 ☆Squid 身份认证

用户/密码认证为 Squid 管理提供了更多便利，最常用的认证方式是 NCSA。从 Squid 2.5 版本开始，NCSA 认证包含在了 basic 中，而非以前单独的认证模块。下面来看看实现认证的具体操作。
首先在编译时配置选项应包括以下配置:
--enable-auth="basic" --enable-basic-auth-helpers="NCSA"

然后需要借助 Apache 的密码管理程序 htpasswd 来生成用户名/密码对应的文件
htpasswd -c /etc/squid/passwd panda
只在第一个生成文件的时候用-c,以后用-c 会覆盖

外部认证程序在/usr/lib/squid 下
Squid.conf 中，包括以下几个相关选项:
#该选项指出了认证方式（basic)、需要的程序（ncsa_auth）和对应的密码文件（password）
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd
指定认证程序的进程数
auth_param basic children 5
浏览器显示输入用户/密码对话框时的领域内容 为 My Proxy Caching Domain
auth_param basic realm My Proxy Caching Domain
基本的认证有效时间 ,过了有效时间还要继续使用需要重新输入用户名和密码
auth_param basic credentialsttl 2 hours

普通用户需要通过认证才能访问 Internet 允许经过认证的用户访问
acl auth_user proxy_auth

http_access allow auth_user

通过以上的配置即可完成认证工作。有的读者可能要问:认证只针对普通用户，而高级用户是直接上网的，该怎么处理呢？其实，这两种用户是可以共存的。如前所述，Squid 是顺序处理 http_access 的，所以在 http_access 处理过程中，如果先处理 normal 用户，那么当前用户无论是否属于高级用户，都会被要求进行认证;相反如果先处理高级用户，剩下的就只有需要认证的普通用户了。例如以下配置代码:
http_access allow normal (需要认证)
http_access allow advance （不需要认证）

不管是否为 noauth 用户，都要求进行用户名/密码验证。正确的方法是将二者位置交换，代码如下:
http_access allow advance
http_access allow normal
这时，高级用户不会受到任何影响。

600 ☆Squid 初始化

squid -z
建立缓冲区目录结构
/var/spool/squid 下

601 ☆Squid 设置错误提示为中文

/etc/squid/errors -> /usr/share/squid/errors/English
改为
ln -s /etc/squid/errors /usr/share/squid/errors/Simplify_Chinese/

602 ☆Squid 日志

/var/log/squid/access.log

603 ☆squid 实验

```
# 服务器配置
http_port 192.168.152.128:3128
cache_mgr root@panda.com
cache_dir null /tmp
cache_access_log /var/squid/access.log
cache_log /var/squid/cache.log
cache_store_log /var/squid/store.log
visible_hostname No1.proxy
client_mask 255.255.255.255
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_user_host_header on

# 用户分类
acl advance arp 00:01:02:1f:2c:3e 00:01:02:3c:1a:8b ...
acl normal proxy_auth REQUIED
acl all src 0.0.0.0

# 行为分类
acl mmxfile urlpath_regex \.mp3$ \.avi$ \.exe$
acl conncount maxconn 3
acl worktime time MTWHF 8:30-12:00 14:00-18:00
```

```
acl sinapage dstdomain ok.sina.com.cn
acl qq dstdomain .tcccent.com.cn

# 处理
http_access allow advance
http_access deny conncount normal
http_access deny !worktime
http_access deny mmxfile
http_access deny sinapage
http_access deny qq
http_access allow normal
```

配置后的状况是，advance 组可以不受任何限制地访问 Internet，而 normal 组则只能在工作时间上网，而且不能下载多媒体文件，不能访问某些特定的站点，而且发送请求不能超过 3 个。

604 ☆邮件系统

电子邮件系统的工作原理
电子邮件系统的运作方式与其它的网络应用有着根本上的不同。在其它的绝大多数的网络应用中，网络协议直接负责将数据发送到目的地。而在电子邮件系统中，发送者并不等待发送工作完成，而是仅仅将要发送的内容发送出去。

电子邮件的协议标准是 TCP/IP 协议族的一部分。它规定了电子邮件的格式和在邮局间交换电子邮件的协议。

每个电子邮件都分为两部分:邮件头和邮件内容。TCP/IP 对电子邮件的邮件头的格式作了确切的规定，而将邮件内容的格式让用户自定义。在邮件头中最重要的两个组成部分就是发送者和接收者的电子邮件地址。

电子邮件地址的格式如下:
用户名@电子邮局域名 例:abc@990.net

电子邮件的传输协议(也就是在邮局间交换电子邮件的协议)主要有 SMTP(简单邮件传输协议)、POP(电子邮局协议)，以及现在新兴的 IMAP(互联网邮件应用协议)。

整个电子邮件应用系统由两大部分构成:
1. 电子邮局系统;
电子邮局行使着像传统邮局的功能，它在发送者和接收者之间起着一个桥梁作用。它是运行在电子邮局服务器上的一个服务器端程序。最常用的有 Microsoft 的 IIS 和 sendmail 等。
2. 电子邮件发送、接收系统。

电子邮件发送、接收系统则象遍及千家万户的邮箱，发送者和接收者通过它将邮件从电脑中发送和接收邮件。这个部分是一个运行在电脑中的客户端程序，最常用的有 Microsoft 的 Outlook Express，Netscape，The Bat，Foxmail，方正飞扬等。

有 3 种可提供的邮件服务

Message Transfer Agent
MTA
消息传输代理(发送邮件服务器)

Mail Delivery Agents
MDA
邮件传递代理(邮件处理器)

Mail User Agent
MUA
邮件用户代理(邮件客户端上运行的程序)

MTA,Mail transfer agent sendmail, postfix, qmail
MUA,Mail user agent mail, Mozilla, elm
MDA,Mail delivery agent procmail, maildrop

605 ☆邮件服务器

邮件服务器是电子邮件系统的核心构件，它的主要功能是发送和接收邮件，同时向发件人报告邮件的传送情况。根据用途的不同，可以将邮件服务器分为发送邮件服务器(SMTP 服务器)和接收邮件服务器(POP3 服务器或 IMAP4 服务器)。

sendmail 和 postfix。您可以选择任何一个 MTA

Sendmail 并不处理最终的投递，当然也不会处理如何把邮件提交给最终用户这样的任务。一般来说，我们总是在 Windows 客户机器上处理各种电子邮件，因此需要一个服务程序负责将 sendmail 存储的邮件转交给 Windows 或其他任何客户机器。有两种基本的方法，一种是将邮件传送到客户的本地机器上处理，这是通过所谓的邮局协议实现的;另一种是允许用户远程操作其邮箱并且实现对邮件的浏览和管理，这是通过所谓的 IMAP 协议。

606 ☆SMTP 协议

SMTP 即简单邮件传输协议，它是一组用于由源地址到目的地址传送邮件的规则，由它来控制信件的中转方式。SMTP 协议属于 TCP/IP 协议簇，它帮助每台计算机在发送或中转信件时找到下一个目的地。通过 SMTP 协议所指定的服务器，就可以把 Email 寄到收件人的服务器上了。SMTP 服务器则是遵循 SMTP 协议的发送邮件服务器，用来发送或中转发出的电子邮件。

607 ☆POP3 协议

POP3 即邮局协议的第 3 个版本，它规定怎样将个人计算机连接到 Internet 的邮件服务器和下载电子邮件的协议。它是 Internet 电子邮件的第一个离线协议标准，POP3 允许从服务器上把邮件存储到本地主机即自己的计算机上，同时删除保存在邮件服务器上的邮件。遵循 POP3 协议来接收电子邮件的服务器是 POP3 服务器。

608 ☆IMAP4 协议

IMAP4 即 Internet 信息访问协议的第 4 个版本，是用于从本地服务器上访问电子邮件的协议，它是一个客户/服务器模型协议，用户的电子邮件由服务器负责接收保存，用户可以通过浏览信件头来决定是否要下载此信。用户也可以在服务器上创建或更改文件夹或邮箱，删除信件或检索信件的特定部分。

609 ☆POP 和 IMAP 差异

虽然 POP 和 IMAP 都是处理接收邮件的，但两者在机制上却有所不同。在用户访问电子邮件时，IMAP4 需要持续访问服务器，POP3 则是将信件保存在服务器上，当用户阅读信件时，所有内容都会被立即下载到用户的机器上。因此，可以把 IMAP4 看成是一个远程文件服务器，而把 POP 看成是一个存储转发服务器。就目前情况看，POP3 的应用远比 IMAP4 广泛得多。

610 ☆mail

使用权限:所有使用者

使用方式:mail [-iInv] [-s subject] [-c cc-addr] [-b bcc-addr] user1 [user 2 ...]

说明:
mail 不仅只是一个指令， mail 还是一个电子邮件程式，不过利用 mail 来读信的人应该很少吧！对于系统管理者来说 mail 就很有用，因为管理者可以用 mail 写成 script ，定期寄一些备忘录提醒系统的使用者。

参数:
i 忽略 tty 的中断讯号。 (interrupt)
I 强迫设成互动模式。 (Interactive)
v 列印出讯息，例如送信的地点、状态等等。 (verbose)
n 不读入 mail.rc 设定档。
s 邮件标题。
c cc 邮件地址。
b bcc 邮件地址。

将信件送给一个或以上的电子邮件地址，由于没有加入其他的选项，使用者必须输入标题与信件的内容等。而 user2 没有主机位置，就会送给邮件伺服器的 user2 使用者。
mail user1@email.address
mail user1@email.address user2

将 mail.txt 的内容寄给 user2 同时 cc 给 user1 。如果将这一行指令设成 crontab 就可以定时将备忘录寄给系统使用者。
mail -s 标题 -c user1 user2 < mail.txt

mail 查看/var/spool/mail/目录下自己邮箱内容(每个用户会有一个文件保存自己所有文件)。以 q 退出把看过的邮件保存在~/mbox 文件中。
mail test@panda.com 直接发邮件给人。
mail -s 'title text' test@panda.com < mail.txt 把文档中内容邮寄出去。
mail -f ~/mbox 查看 home 目录下邮箱内容。

611 ☆uuencode/uudecode

用 mail 发附件也是可行，要用到 uuencode and uudecode 命令进行编码。

编码:uuencode [file] name
uuencode hello >hello.uue
default input is stdin;
default output is stdout.

解码:uudecode [-o outfile] name
uudecode hello.uue
可以用-o 选项输出另外一个文件名。
uuencode ~/.bashrc -o bashrc | mail -s 'test uuencode' test@panda.com

612 ☆system-switch-mail

切换 sendmail 和 postfix

613 ☆sendmail

sendmail 提供 SMTP 服务

614 ☆sendmail 安装

如果你在安装 LINUX 的时候，选择了 E-MAIL 服务，sendmail 就已经安装在 LINUX 系统中了，并且已经作了一些最基本的设置。
sendmail-8.9.3-10.i386.rpm sendmail 可执行文件

sendmail-cf-8.9.3-10.i386.rpm sendmail.cf 生成器
sendmail-doc-8.9.3-10.i386.rpm sendmail.cf 文档
还有 m4 和 procmail

615 ☆sendmail 使用前配置

sendmail 的 DNS 设置
当 sendmail 程序得到一封待发送的邮件的时候，它需要根据目标地址确定将信件投递给那一个服务器，这是通过 DNS 服务实现的。例如，有一封邮件的目标地址是 panda@panda.com，那么，sendmail 首先确定这个地址是用户名(panda)+机器名(panda.com)的格式，然后，通过查询 DNS 来确定需要把信件投递给某个服务器。

先用 ifconfig 查看服务器的 ip

再把 ip 写入/etc/resolv.conf
; generated by /sbin/dhclient-script
search panda.com
nameserver 192.168.152.128

确认/etc/hosts
Do not remove the following line, or various programs
that require network functionality will fail.
127.0.0.1 panda.panda.com panda panda

然后在/etc/named.conf 中加入
zone "panda.com" IN {
type master;
file "panda.com.zone";
};

最后生成/var/named/chroot/var/named/panda.com.zone
$TTL 86400
@ IN SOA panda.panda.com. root.panda.panda.com. (
2006062700 ; Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400) ; Minimum
IN NS panda
IN MX 10 panda
panda IN A 192.168.152.128

重启 dns
service named restart

DNS 数据中，与电子邮件相关的是 MX 记录，这可以在查询 DNS 时设置查询类型为 mx 来得 到:
[root@panda ~]# nslookup
> panda.panda.com
Server: 192.168.152.128
Address: 192.168.152.128#53

Name: panda.panda.com
Address: 192.168.152.128

> set q=mx
> panda.com
Server: 192.168.152.128
Address: 192.168.152.128#53

panda.com mail exchanger = 10 panda.panda.com.

在一般的情况下，mx 交换器会自动把信件内容转交给目标主机，不过，也存在这样的情况 ，目标主机(比如 panda.com)可能并不存在，或者不执行 smtp 服务，而是由其 mx 交换器来执行信件的管理，这时候，最终的信件将保存在 mx 机器上，直到用户来察看它。

如果 DNS 查询无法找出对某个地址的 MX 记录(通常因为对方没有信件交换主机)，那么 sendmail 将是试图直接与对方的主机(来自邮件地址)对话并且发送邮件。例如,test@china.com 在 DNS 中没有对应的 MX 记录，因此 sendmail 在确定 MX 交换器失败后，将从 DNS 取得对方的 IP 地址并直接和对方对话试图发送邮件。

sendmail 发送邮件时，如果经过设定的时间后仍然未能将信件投递到目的主机，它将返回一个错误信息并且休息一段时间，然后重新试图投递，如果连续多次失败，sendmail 最终将放弃投递并将错误信息投递给 postmaster 用户。在许多机器上，postmaster 用户是 root 用户的一个别名(参考下面关于别名的内容)，你应该将它设置为邮件的实际管理员的用户名。

例如，你可能会用 mail 程序向某个地址这样发信:
$mail someone@somedomain.com
To:other@otherdomain.com
Subject:test mail
test
.

someone@somedomain.com 必须准确到某台主机上

那么，当 sendmail 发信的时候，它是向 someone@somedomain.com 发信而不是 other@otherdomain.com。相应地，如果你想向两个人发信，例如你在 outlook 里面写上:"投递给 user1@a.com，抄送 user2@b.com"，那么 sendmail 应该怎么做？直接同时向两个地址发信吗？否，它试图构造两个包装(称为信封)，每个包装上只列出一个投递地址，各投递一次。虽然邮件正文的头部仍然包含两个地址，但是 sendmail 不会看它。

配置文件
/etc/sysconfig/sendmail
DAEMON=yes
QUEUE=1h
每一个小时传递队列中的文件

616 ☆sendmail 防火墙

iptables -I INPUT -p tcp --dport 25 -j ACCEPT

617 ☆sendmail 启动

启动用
service sendmail restart
或
sendmail -bd -q30m
该命令以后台进程方式(-bd)运行，并使其每隔 30 分钟(-q30m)轮询一次未发送邮件队列，检查是否有新邮件。-q1h 表示每隔一个小时发送一次邮件

618 ☆sendmail 校验

看 sendmail 是否启动
netstat -nutlp | grep :25

确定 sendmail 没有在启动的时候出现错误
/var/log/maillog 中。检查此文件中的最后出现"starting"的地方以确保 sendmail 在启动的时候没有任何错误。

sendmail 可执行文件位于/usr/sbin/sendmail。为了确定 sendmail 是否正确标识您的主机名称，通过命令行开关开启其调试模式并且设定为 0:
[root@panda mail]# sendmail -d0 < /dev/null
Version 8.13.1
  Compiled with: DNSMAP HESIOD HES_GETMAILHOST LDAPMAP LOG MAP_REGEX
                 MATCHGECOS MILTER MIME7TO8 MIME8TO7 NAMED_BIND NETINET
NETINET6

NETUNIX NEWDB NIS PIPELINING SASLv2 SCANF STARTTLS
TCPWRAPPERS
USERDB USE_LDAP_INIT

========== SYSTEM IDENTITY (after readcf) ==========
(short domain name) $w = panda
(canonical domain name) $j = panda.panda.com
(subdomain name) $m = panda.com
(node name) $k = panda
==

Recipient names must be specified

如果 sendmail 返回您的主机名称为 localhost,您可能错误配置了/etc/hosts 文件。检查您的/etc/hosts 文件，删除所有的但记住留下 localhost 的指向。
127.0.0.1 panda.panda.com panda localhost
如果/etc/hosts 文件是正确的，那么检查一下在/etc/sysconfig/network 中的 HOSTNAME 的定义。

试图向 root@panda 发送简单的邮件。您可以看到一个合理的您的主机的转发服务器的 SMTP 交换。
echo "hello root" | mail -v -s hello root@panda

[root@panda ~]# echo "hello root" | mail -v -s hello root@panda
root@panda... Connecting to [127.0.0.1] via relay...
220 panda.panda.com ESMTP Sendmail 8.13.1/8.13.1; Thu, 27 Jul 2006 08:46:11 +0800
>>> EHLO panda.panda.com
250-panda.panda.com Hello panda.panda.com [127.0.0.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5
250-DELIVERBY
250 HELP
>>> MAIL From:<root@panda.panda.com> SIZE=42 AUTH=root@panda.panda.com
250 2.1.0 <root@panda.panda.com>... Sender ok
>>> RCPT To:<root@panda.panda.com>
>>> DATA
250 2.1.5 <root@panda.panda.com>... Recipient ok
354 Enter mail, end with "." on a line by itself

>>> .
250 2.0.0 k6R0kBQO005444 Message accepted for delivery
root@panda... Sent (k6R0kBQO005444 Message accepted for delivery)
Closing connection to [127.0.0.1]
>>> QUIT
221 2.0.0 panda.panda.com closing connection

You have mail in /var/spool/mail/root

如果 SMTP 交换向上面一样正确，那么消息将被转发到您的工作站上的本地的转发服务器上，并且 mailq -Ac 将会报告一个空的对列。接下来检查 mail(不使用参数)来检查一下消息是否从本地的转发到 server1。这样对列也应该是空的。
[root@panda public_html]# mailq -Ac
/var/spool/clientmqueue is empty
                Total requests: 0

您的消息是不是在/var/log/maillog 中正确的记录呢？在下面的步骤中，监视文件/var/log/maillog。下面的命令将会十分的有用:

xterm -e tail -f /var/log/maillog &

619 ☆sendmail 收取邮件

为了安全的原因，sendmail 和 postfix 的缺省的配置允许发邮件但是不允许从网络上接收邮件(缺省的它们只接受从回环接口上的连接)。
确保在 Server 上的 sendmail.mc 文件中的 DAEMON_OPTIONS 被注释并且重新编译 sendmail.cf 文件使得能构接受来自其他主机的电子邮件。

修改 /etc/mail/sendmail.mc
使用 dnl 注释在下面的行之前，就象这样:
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')

先将您的 sendmail.cf 文件做一个备份:
cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.bak

在同一个目录下，编译 sendmail.cf
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

重新启动 sendmail

620 ☆sendmail 别名

sendmail 的主要的别名配置文件是/etc/aliases。
在 sendmail 决定消息的接受者的目的地的之前，其先试图在别名中查找。
为了优化查找，sendmail 为其别名记录建立了一个哈希表数据库/etc/aliases.db 该文件通过 newaliases 命令产生(该命令是 sendmail -bi 的同名)

下列命令将增加用户 student(如果不存在的话)
useradd student

在/etc/aliases 行加入如下的行:
me: student(别名:实际用户名)
wizards: root, me
methere: student@panda.panda.com

现在运行
newaliases
命令来更新数据库

尝试发送邮件给您定义的收件人:

echo "hello there" | mail -s "hello" me
echo "hello there" | mail -s "hello" wizards
echo "hello there" | mail -s "hello" methere

您是否得到了期望的结果？是否所有的位于 wizards 的收件人都受到了邮件？如果没有，su - 到不是 root 的用户再试一次。

如果没有收到,可能需要在/etc/mail/access 中打开权限

621 ☆sendmail 允许转发

通过控制您的机器的混杂转发，您可以使得任何人都能够将您的机器作为转发的主机。
配置/etc/mail/sendmail.mc, 通过加入如下行使得 m4 前置处理器允许混杂转发(可以发到任何地址):

/etc/mail/sendmail.mc 中添加
FEATURE(`promiscuous_relay')dnl

使用 m4 前置处理机通过这个模板文件生成一个新的 sendmail 配置文件，然后将新生成的文件与通过 sendmail RPM 软件包提供的进行比较
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.relay

diff /etc/mail/sendmail.relay /etc/mail/sendmail.cf

使用混杂转发以后会有多大的不同呢？现在将新建立的 sendmail.relay 放置在恰当的位置上，重新启动 sendmail.
mv /etc/mail/sendmail.cf /etc/mail/sendmail.cf.accept-mail
cp /etc/mail/sendmail.relay /etc/mail/sendmail.cf
service sendmail restart

让您的伙伴扮演恶意的垃圾邮件的发送者，该人能够通过 telnet 到您的机器上的 smtp(sendmail)的 25 号断口，进行垃圾邮件发送地址的欺骗，在 panda 主机上键入如下命令:

[root@panda ~]# telnet panda 25
Trying 127.0.0.1...
Connected to panda.panda.com (127.0.0.1).
Escape character is '^]'.
220 panda.panda.com ESMTP Sendmail 8.13.1/8.13.1; Thu, 27 Jul 2006 01:12:37 +0800
helo panda.panda.com
250 panda.panda.com Hello panda.panda.com [127.0.0.1], pleased to meet you
mail from: panda@panda.panda.com
250 2.1.0 panda@panda.panda.com... Sender ok
rcpt to:root@panda.panda.com
250 2.1.5 root@panda.panda.com... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Subject: Faked
this was faked!
.
250 2.0.0 k6QHCb8x007709 Message accepted for delivery
quit
221 2.0.0 panda.panda.com closing connection
Connection closed by foreign host.

垃圾邮件现在送到您的机器上了。下一步，发给其他主机

622 ☆sendmail 不允许转发

通过替换新的 sendmail.cf 为接受传入的信件的配置文件来恢复缺省的 sendmail 的配置，并且重新启动 sendmail:
注释掉这行
FEATURE(promiscuous_relay)dnl

mv /etc/mail/sendmail.cf.accept-mail /etc/mail/sendmail.cf

service sendmail restart

让您的伙伴再从 stationY 转发垃圾邮件。您的 sendmail 还是一个转发器么？任何一个转发的都会产生如下的消息:

550 5.7.1 kddk@kdk.com... Relaying denied

但是本机是可以转发的(access 文件中定义)

623 ☆sendmail 选择性的转发

对于特定的主机，域或者网络，编辑/etc/mail/access 并且重新启动 sendmail。
为了允许所有在 panda.com 域中的机器可以把您的机器作为邮件转发服务器，你在/etc/mail/access 中添加如 panda.com 域。

然后用
makemap hash access.db<access

624 ☆sendmail 配置文件

-rw-r--r-- 1 root root 1637 7 月 30 08:09 /etc/aliases
-rw-r----- 1 root smmsp 12288 8 月 21 01:44 /etc/aliases.db

-rw-r--r-- 1 root root 331 2004-09-01 access
-rw-r----- 1 root root 12288 8 月 21 01:44 access.db
-rw-r--r-- 1 root root 0 2004-09-01 domaintable
-rw-r----- 1 root root 12288 7 月 27 23:41 domaintable.db
-rw-r--r-- 1 root root 5588 2004-09-01 helpfile
-rw-r--r-- 1 root root 64 2004-09-01 local-host-names
-rw-r--r-- 1 root root 0 2004-09-01 mailertable
-rw-r----- 1 root root 12288 7 月 27 23:41 mailertable.db
-rw-r--r-- 1 root root 1035 2004-09-01 Makefile
-rw-r--r-- 1 root root 55561 8 月 21 01:44 sendmail.cf
-rw-r--r-- 1 root root 6790 8 月 21 01:44 sendmail.mc
drwxr-xr-x 2 root root 4096 7 月 27 23:44 spamassassin
-rw-r--r-- 1 root root 40283 7 月 27 23:51 submit.cf
-rw-r--r-- 1 root root 952 2004-09-01 submit.mc
-rw-r--r-- 1 root root 127 2004-09-01 trusted-users
-rw-r--r-- 1 root root 0 2004-09-01 virtusertable
-rw-r----- 1 root root 12288 7 月 27 23:41 virtusertable.db

625☆m4

sendmail 是一个极为复杂的程序，其行为主要地依赖于在 UNIX 界"臭名昭著"的/etc/sendmail.cf 配置文件。实际上，我怀疑会有谁真的从头去写一个 sendmail.cf 文件。一般来说，我们总是用 m4 宏处理来书写 sendmail.cf。实际上，m4 程序几乎和 sendmail.cf 一样复杂，不过，通常我们只需要关心一些比较重要的部分。

要使用宏处理程序，必须确定你已经安装了 m4 和 sendmail-cf 这样两个软件包

用 m4 程序可以生成一个 sendmail.cf:
m4 sendmail.mc > /sendmail.cf

626☆mc 语法

一个 mc 模板文件通常可以包含几个段落:
注意里面的正反引号，而且不能随便加入空格。

sendmail 宏定义说明
divert(n) 为 m4 定义一个缓冲动作，当 n=-1 时缓冲被删除，n=0 时开始一个新缓冲
OSTYPE 定义宏所使用的操作系统，该宏允许 m4 程序增加同相关操作系统相关的文件
Domain 定义 MTA 将使用哪些域来传输邮件
Feature 定义配置文件中使用的一个特定的功能集
Define 定义配置文件中的一个特定的选项值
MASQUERADE_AS 定义 sendmail 来应答邮件的其它主机名
MAILER 定义 sendmail 使用的邮件传输方法
dnl 注释

divert
通常总是设置为 divert(-1)让 m4 在输出中去掉一些垃圾。

OSTYPE OSTYPE
定义使用的操作系统类型，当然在我们的情况下就是 linux，但是一定要注意 m4 程序中引号的用法，一个反引号和一个正引号才代表把对应的东西括起来。

define
定义一些全局设置，对于 Linux 系统，设置了 OSTYPE 之后，可以定义下面的一些全局 参数，如果不定义，就使用缺省值。例如:

define(ALIAS_FILE，/etc/aliases)
变量名 说明(方括号中为缺省值)

ALIAS_FILE [/etc/aliases]

别名文件的位置。如果有多个别名文件，需要把它们用引号括起来(别忘了引号规则 ！)。

confCR_FILE [/etc/mail/relay-domains]
缺省的域定义文件，在这个域中定义的域中机器可以通过你的服务器进行邮件发送。

HELP_FILE [/usr/lib/sendmail.hf]
此文件中含有对 SMTP 的 HELP 命令进行响应时要列出的信息。

QUEUE_DIR [/var/spool/mqueue]
邮件队列文件所在目录。

STATUS_FILE [/etc/sendmail.st]
sendmail 的状态信息文件。

LOCAL_MAILER_PATH [/bin/mail]
用于投递本地邮件的程序。

LOCAL_MAILER_FLAGS [rmn9]
local mailer 要用到的标志，永远包含标志 lsDFM。

LOCAL_MAILER_ARGS [mail -d $u]
在投递本地邮件时所传送的参数。

LOCAL_MAILER_MAX [没有]
如定义了此参数，则为此邮件服务器所能接收最大单个邮件大小。

LOCAL_MAILER_CHARSET [没有]
如果定义了此参数，则被转化为 MIME 格式的从其他地址到 local mailer 的含有 8 位字符 的信息将被标为此字符集。

LOCAL_SHELl_PATH [/bin/sh]
用于投递利用管道功能处理的邮件的 shell.

LOCAL_SHELL_FLAGS [eu9]
prog mailer 用到的标志。在此标志中永远包含标志 lsDFM.

LOCAL_SHELL_DIR [$z:/]
shell 运行时所要查找的目录路径。

USENET_MAILER_PATH [/usr/lib/news/inews]
用于投递电子新闻组的程序名称。

USENET_MAILER_FLAGS [rlsDFMmn]
usenet mailer 的投递标志。

USENET_MAILER_ARGS [-m -h -n]
usenet mailer 的命令行参数。

USENET_MAILER_MAX [100000]
usenet mailer 所能接收的最大信息大小。

SMTP_MAILER_FLAGS [没有]
SMTP mailer 附加标志。对所有基于 SMTPmailer 其默认标志为 mDFMUX;基于 esmtp 的邮差(mailer)加上 a 标志;而基于"smtp8"的邮差则加上 8。

SMTP_MAILER_MAX [没有]
使用 smtp,smtp8 或 esmtp 传输的单个邮件最大容量。

SMTP_MAILER_CHARSET [没有]
如果定义了此参数，则被转化为 MIME 格式的从其他地址到任一个 smtp mailer 的含有 8 位字符的信息将被标为此字符集。

POP_MAILER_PATH [/usr/lib/mh/spop]
pop 邮差的路径名。

POP_MAILER_FLAGS [Penu]
pop 邮差附加标志。同时总是加上标志 lsDFM。

POP_MAILER_ARGS [pop $u]
传给 pop 邮差的参数。

PROCMAIL_MAILER_PATH [/usr/local/bin/procmail]
procmail 程序的路径名。此外 FEATURE(local procmail)也用到此参数。

PROCMAIL_MAILER_FlAGS [SPhnu9]加给 Procmail 邮差的标志。同时总是加上"DFM"标志 。

PROCMAIL_MAILER_MAX [没有]
procmail 所接收的最大单个邮件容量。如果你对某些人发送巨大的邮件感到困扰，启 用这个选项。

FEATURE
定义 sednamil 的一些运行参数，通常对我们来说最重要的一些选项是:

use_cw_file

读取文件/etc/sendmail.cw 以确定这台机器应该替哪些机器接受邮件。此主机的别名 。当你使用 MX 记录将此主机定义为其他主机的邮件交换机时需要使用这个特性。例如:
FEATURE(use_cw_file)

relay_hosts_only
通常情况下，sendmail 为 sendmail.cf 中明确列出的域(一般是 localhost)和/etc/ mail/relay-domains 中定义的域进行投递代理。缺省下这两处定义的都是域的名字。如果你定义了这个参数，那么这两处的内容将被解释为主机名字。

use_ct_file
读取文件/etc/sendmail.ct 以取得系统"信任"的用户名字，这些用户可以使用-f 设 置其发信信封上的 from 地址而不产生警告信息。

redirect
使用 REDIRECT 特性，这个特性允许你对某些已经搬迁的用户发出重定向信息。(见下 一节)。例如:
FEATURE(redirect)

mailertable
包含一个用于覆盖到特定域路由(routing)的"mailer table".此特性参数定义可以是 一个关键词定义。如未指定任何参数， 其定义通常是:
FEATURE(mailertable,`hash -o /etc/mailertable)

domaintable
包含一个用于提供域名映象的"domain table"，当改变你自已的域名时可能有用(如 你公司由 oldname.com 改为 newname.com)。其定义通常是:
FEATURE(domiaintable，`hash -o /etc/domaintable)

always_add_domain
在本地发送邮件时也加上其主机域名。例如:
FEATURE(always_add_domain)

allmasquerade
如果使用了伪装(masquerading,使用 MASQUERADE_AS),则此特性将使接收者的地址也伪 装为来自所伪装为主机。

limitd_masquerade
通常情况下$w 所列出的所有主机将被伪装。如果使用了此特性，则只对那些$m 所列出 的主机进行伪装。

masquerade_entire_domain

如使用了伪装且设置了 MASQUERADE_DOMAIN,此特性将引起 地址重写，使所要伪装的网 域整个被隐藏。所有含有被伪装域名的主机用伪装域名(通常是 MASQUERADE_AS)进行重写 。

masquerade_envelope
用此特性告知 sendmail 将信封和信件头中上的发送者和接收者进行伪装。

定义了有关 masq 的选项之后，就可以使用伪装了，可以直接将伪装命令写入 mc 模板， 示例如下:
MASQUERADE_AS(masq.com)
MASQUERADE_DOMAIN(foo.org)
这意味着我们的 someone@foo.org 发信的时候，sendmail 将会把它的信封伪装为 some one@masq.com。这对于统一整个域的电子邮件是非常重要的。

virtusertable
允许在同一个主机上使用多个虚拟域。参考下一节。例如:
FEATURE(`virtusertable,`hash -o /etc/mail/virtusertable)

nullclient
这是一个特殊情况--它生成一个除了支持将所有的邮件通过本地的基于 SMTP 的网络转 递到一个中心邮件 HUB 之外不含任何内容的配置文件。其参数是此邮件 HUB 的主机名。唯 一可与 nullclient 一起使用的其他特性是"nocanonify"(这样可以使非完全地址可通过 S MTP 连接进行发送; 通常情况下地址将使用伪装名字转变为完全邮件名称，此伪装名字默 认值为邮件 HUB 主机的名字)。 在此特性使用是不应定义任何邮差。当然也不进行别名 处理或转寄。

local_procmail
使用 procmail 作为本地邮差。

smrsh
对到程序的邮件使用使用 sendmail 发行版所带的 SendMail Restricted SHell (smrsh)而不是/bin/sh。由于 sendmail 是以 root 权限执行，某个发送到恶意程序的邮件可以破 坏系统，只要利用别名转向使得邮件被转发到对应的程序，因此缺省下 sendmail 用 smrs h 来处理邮件转发到程序的请求。这可以提高本地系统管理员控制对那些通过邮件运行程 序的行为，例如

FEATURE(`smrsh,`/usr/sbin/smrsh)
注意有些程序无法通过 smrsh 运行(例如 majordomo 的 wrapper 程序)，这是出于安全 性的考虑，smrsh 不准用户程序使用一些 setuid 功能。如果你一定要使用这些程序，清将 smrsh 定义成其他 shell 程序，如 sh。

access_db
本地存取控制文件的名字，缺省是/etc/mail/access.db，也可以用命令行指出，例 如:

FEATURE(`access_db)

或者 或者
FEATURE(`access_db,`hash -o /etc/mail/access)

mailertable
允许使用 mailertable 文件。这个文件定义对某确定的域使用什么样的邮差。例如:
FEATURE(`mailertable'，`hash -o /etc/mail/mailertable)。

blacklist_recipients
允许你用前面定义的 access_db 来禁止某个地方来的邮件，或是某个人的邮件，等等 。

relay_based_on_MX
是否允许别人用你的机器当成 MX 交换器。如果你设置了这个选项，那么任何人只要在 域名服务器中将你的机器设置成为他的 MX 交换就可以用你的机器转发电子邮件。这个功 能意味着:你的机器替它接受电子邮件，再提交给它;一般来说这个功能是不必要的; 如果你一定要使用这个选项，记住你可能被庞大的邮件流量吞没。但是在一种情况下这 个功能又是不可缺少的:假如你的系统有防火墙，只有邮件服务器能够对外连接，那么 这个功能是使网络内部其他主机能够接受自己电子邮件的唯一方法。

DOMAIN
这个关键字一般用来定义邮件中继，假如你的系统里面除了 Internet 互连之外还有类 似 Decnet，UUCP 之类的东西，那么你就需要设置 DOMAIN 来保证非 internet 的邮件被正确 中继。对于一般的系统，不需要定义这个属性。

MAILER
定义可以使用的投递程序(邮差)。
例如:
MAILER(smtp)
定义 smtp 投递。

MAILER(local)
定义局部投递。

如果你想做邮件服务，这样两个邮差是必须的。

627 ☆sendmail.mc

628 ☆sendmail.cf

m4 sendmail.mc > /etc/sendmail.cf

629☆access

限制某些域的邮件
/etc/mail/access 访问控制列表设置文件的格式是这样:
[地址] [操作]
中间的分割符是空格键。

[地址]栏可以是主机地址或者名字，也可以是统配符，规则是这样:
yourdomain.com 代表所有*.yourdomain.com 的名字。
192.168.12 代表所有 192.168.12.*的地址。
202.135 代表所有 202.135.*.*的地址。
someone@somedomain.com 代表一个特定的邮件发信人

[操作]栏通常有:
OK 正常接受这封邮件,远程主机可以向你的邮件服务器发送邮件;
RELAY 允许 SMTP 代理投递，这样这封邮件就可以从你的机器中转到别的机器上去
REJECT 拒绝接受,不能向你的邮件服务器发邮件和不能中转;
DISCARD 发来的邮件将被丢弃，同时并不向发送者返回错误信息。忽略这封邮件，这种情况下，邮件看上去是正常投递了，但是由于没有人接受，邮件会自动地"消失"在网络中。
nnn text-- 发来的邮件将被丢弃，但 sendmail 将会向发送者返回 nnn 确定的 smtp 代码和 text 变量确定的文本描述。

错误代码+任何其他字符串:将向发信者返回这个字符串作为出错信息。错误代码是 RFC 822 定义的标准出错代码。例如
550 We don like a spammer!
客户机器在投递邮件的时候，就会产生一个"we don like a spammer"投递失败信 息。比如，你认为 someone@spammer 是个专门投递垃圾邮件的家伙，那么你可以这样写:
someone@spammer 550 we don like a spammer

然后重新启动 sendmail 就可以使用这些功能了。

192.168 RELAY
panda.NET OK
panda.COM REJECT
panda.COM 550 SORRY,WE DON'T ALLOW SPAMMERS HERE
panda.ORG DISCARD

设置完成后要用 makemap hash access.db < access 命令生成数据库。

630☆access.db

一般/etc/mail/access.db 是一个散列表数据库，它是用/etc/mail/access 为模版产生出来的。

631☆aliases

/etc/aliases，这个文件用来设置用户的别名。
文件/etc/aliases 允许为本地用户，应用程序，甚至其他别名提供虚拟邮箱:

aliases 文件的格式是
邮件别名:实际用户名

如果一个别名有多个用户就用逗号分开 ，每个别名一行。

1.最简单的情况是需要作信件分发的情况。

要把发给 postmaster 的信件发送给 supervisor 和 manager，需要写上这样一行:
postmaster:supervisor,manager

别名还可以用在这样的情况，即定义自动的邮件转发。

某个用户以前在你的系统上接受电子邮件，现在他有了一个新的电子邮件，希望发到你的机器上的邮件自动被转发到他新的电子邮件地址上，那么，可以使用类似这样的别名方式:(假设你的机器是 panda@panda.com)
panda:epeppanda@hotmail.com
以后发给 panda@panda.com 的电子邮件就自动中转到 epeppanda@hotmail.com。注意左边自动加上你的机器名字，所以左边只能是账号名字，不能是全限定邮件地址。

2.别名的右侧也可以是文件或程序。

上面的 postmaster 别名可以用这样方法来设置:
panda: :include: /etc/mail/myaliases
include:关键字表示让 sendmail 去读取对应的包含文件。而/etc/mail/myaliases 的内容要设置成:
pandeng
manager

3.要把邮件重定向到程序，可以使用管道

panda:"|/home/panda/testpg"
那么，sendmail 会将发给 panda 的邮件的内容作为/home/panda/testpg 程序的输入来执行这个程序。

另一个常用的办法是重定向。如果你在模板文件中定义了 REDIRECT 特性，那么可以使用这个功能。

某个人在你的机器上开了一个账户 user1，后来迁移到 user2@server2.com。那么，你可以将其别名写成
user1: user2@server2.com.REDIRECT

以后当有人向这个地址发信的时候，你的 sendmail 会将其退回，并且返回一个 551 User not local; please try user2@server2.com 的信息。

在使用别名的时候，必须注意的是不要造成循环，例如 user1 转发给 user2，user2 又将 其转发给 user1....如此循环。在这种情况下，转发 17 次后，sendmail 将把它退还给发信 人。最常见的错误发生在你试图在转发邮件的同时在本地保留备份的情况下，例如:
user1: user1,user2
就构成了一个循环。

在修改了别名文件之后，重新初始化别名数据库:
[root@mail mail]# newaliases
/etc/aliases: 17 aliases, longest 31 bytes, 241 bytes total

也可以使用 sendmail -bi 命令:
[root@mail mail]# sendmail -bi
/etc/aliases: 17 aliases, longest 31 bytes, 241 bytes total
两种方式实际是完全一样的。

当前版本的 sendmail 对各种附加文件和配置文件的属性都有着严格的要求，特别是/etc/aliases 文件，必须至少为 0644 以避免非授权的修改。

632☆.forward

~/.forward
其实该文档的作用和 aliases 数据库的作差不多啦，都是配置别名，做邮件转发的。因为 alises 只能由管理员控制，个人用户不能修改，所以就可以在 test 个人的目录下建立一个转寄文档。以设置个人的邮件转寄列表。文档格式如下:
test
test1
test2
test3
user2@domain.com
.....

这种技术可以让每个用户自己管理自己的邮件别名。但由于个人用户安全意识差，如果设置不当会有安全漏洞，不建议使用。

633 ☆local-host-names

/etc/mail/local-host-names
将 FEATURE(use_cw_file)包含在你的 sendmail.mc 文件当中，Sendmail 将用本地主机名来作为你的本地别名。
主机别名文件
它给出本地主机的别名。如果你的主机有多个名字，或者你的主机是整个域的信件交换主机，你就需要这个文件了。

local-host-names - include all aliases for your machine here.
panda.com
www.panda.com
mail.panda.com
otherdomain.com

如果 Sendmail 没有在收件列表中发现相应的主机名，它将拒绝接受对方发来的邮件。
请记住在修改了这个或其他任何配置文件后你必须重启 Sendmail。

如果你有多个别名或者需要负责的交换域，每个需要单独写上一行。

634 ☆trusted-users

设置可信任用户
可以代表其他用户发邮件的用户

这个可选的特性在 sendmail.mc 中的名称为 use_ct_file。

这个列表中的用户可以改变电子邮件的寄件人。

在/etc/mail/trusted-users 文件中每个用户名占一行。

这对运行在你的服务器上的脚本或应用程序给某个人或组发信息是非常有用的。

它可以更容易的快速识别一个电子邮件的信息。

635 ☆virtusertable

改域名后,可以做域名的转换 |旧域 新域
类似于 aliases 文件,但用于外部用户

/etc/mail/virtusertable

虚拟用户表格向你服务器上的真实邮箱发送虚拟域和邮箱的邮件。这些邮箱可以是本地的，远程的，或是由/etc/aliases 定义的别名。这些是在你的 sendmail.mc 文件中一命令行的形式进行初始配置的。

语法如下
左地址 右地址
中间用 Tab 键分开。

例如:
someone@otherdomain.com localuser

这样一行意味着本来应该发送给 someone@otherdomain.com 的邮件现在要发送给本机的用户 localuser。当然，这意味着:第一，你的 DNS 记录中，本机应该是 otherdomain.com 的 MX 交换器; 第二，你的本机 local-host-names 文件应该包含 otherdomain.com 这个名字。

当然纯粹的这样的域意义不大，但是 sendmail 还支持邮件虚拟域的参数翻译。例如:
@testdomain.com test@mydomain.com
意味着所有发往 xxx@testdomain 的邮件都会被发送到 test@mydomain.com。而
@testdomain.com %1test@mydomain.com
则代表参数转义，例如 user1@testdomain.com 的邮件被发送到 user1test@mydomain.com，user2@testdomain.com 被发送到 user2test@mydomain.com。同样，这样的功能也要通过 MX 记录和 local-host-names 文件加上去。

建立 virtusertable 的方法与建立 access 的办法是一样的:
makemap hash virtusertable.db < virtusertable

然后重新启动 sendmail。

636 ☆virtusertable.db

如同 Apache 一样，sendmail 也允许使用虚拟主机功能，这是通过 FEATURE(virtusertable)功能实现的，而虚拟主机的文件缺省是/etc/mail/virtusertable.db，它用/etc/mail/virtusertable 文件生成，这个文件的形式类似于 aliases 文件，

makemap hash virtusertable.db < virtusertable

637 ☆mailertable

定义邮差
/etc/mail/mailertable 文件(在 FEATURE(`mailertable')里定义)用来定义对某个域名或者用户使用什么样的邮差，如 sz.st-anda.com smtp:[10.100.100.252]等等。一般情况下，并不需要定义这个功能。建立这个文件的方式与上面的几个 hash 数据库相同。

638 ☆spamassassin

给/etc/procmailrc 增加
INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
来支持 SpamAssassin

639 ☆statistics

/var/log/mail/statistics
用 mailstats 来读取这个文件

640 ☆submit.mc

可以让使用 sendmail 的限定于组

641 ☆Makefile

/etc/mail/Makefile
中是编译选项
用 make -C /etc/mail 来处理 mail 目录中的所有文件

642 ☆sendmail 总结

配置接收邮件，/etc/mail/sendmail.mc
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
m4 sendmail.mc > sendmail.cf

添加别名 在/etc/aliases 加入别名,newaliases

不充许转发：默认

转发:在 sendmail.mc 加入
FEATURE（promiscuous_relay) dnl

选择转发: /etc/mail/access

用户和主机访问控制
/etc/mail/access
domain.com RELAY
USER@ REJECT

USER@domain.com REJECT
IP RELAY
makemap hash access.db<access
或
makemap hash access <access .db 可以省略

643 ☆新建 E-Mail 帐号

在 LINUX 中，要为一个新的用户开 E-Mail 帐号是十分简单的。只要在 LINUX 系统中新增一个用户就可以了。那么这个用户帐号和密码就是 E-Mail 的帐号和密码。
我为一个新用户 test 开一个 E-Mail 帐号。就用以下命令:
adduser panda
passwd panda

这样，这个新用户的 E-Mail 地址就是:
panda@panda.com
密码当然就是帐号的密码了。

要防止本地用户利用 sendmail 服务器上的漏洞，最好是让邮件用户只使用电子邮件程序来访问 sendmail 服务器。邮件服务器上的 Shell 帐号不应该被允许，/etc/passwd 文件中的所有用户 shell 都应该被设置为 /sbin/nologin，在添加系统用户时可以 useradd -s /sbin/nologin 用户名，这样添加的用户名就没有了 shell，只能收发信而不能登陆到服务器运行其他程序，安全一点。

644 ☆sendmail 邮件限额

当一个 E-Mail 服务器为许多人提供 E-Mail 服务的时候，无限量的电子邮件将很容易塞满服务器的硬盘，造成很大的负载。如果你的服务器不想为用户提供无限空间的 E-Mail 暂存空间，那么你就可以使用"邮件限额"来给用户一个有限的暂存空间。
其实，它是利用磁盘限额功能来实现的。电子邮件的暂存空间是在/var/spool/mail 目录下，只要通过磁盘限额设定每一个用户在这个目录下能使用的最大空间就可以了。

645 ☆mailstats

邮寄状态查询命令，可查询 sendmail 运行作至今邮件收发总计资料。
M :
msgsfr:发送的邮件数量
bytes_from:邮件容量
megsto:收到邮件的数量。
bytes_to:同上
msgsrej:邮件 deny 的次数。
msgsdis:邮件 discard 的次数。
Mailer :esmtp 对外邮件，local 本地邮件。

646 ☆mailq

邮件队列查询命令。
Q-ID 邮件 id 号。
Size 邮件容量。
Q-Time 邮件进入队列(也就是/var/spool/mqueue 目录)的时间和不能邮寄的原因。
Sender/Recipient 发信和收信人的邮箱地址。

647 ☆sendmail -q

当命令 sendmail -q 发出以后，sendmail 将会试图仍在队列中等待的邮件。
可以在后面跟上时间‘s’是秒,‘m’分钟(缺省),‘h’小时, ‘d’天,‘w’周.

648 ☆sendmail 管理邮件队列

一般当你发送一封邮件的时候，sendmail 倾向于立刻发送这一封电子邮件。但是如果 当前网络忙使得无法立刻投递信件，或者是目标地址的连接速度太慢，无法在短时间内投递到目标地址处，那么 sendmail 将把待发送的邮件排入队列，并在合适的时候重新发送。

队列文件通常存放在/var/spool/mqueue 下面，每个待发送的邮件由几个文件构成
例如，我们可以看到下面的目录文件列表:
[root@mail mqueue]# ls
dfRAA27175 xfAAA00733 xfBAA00819 xfEAA32763 xfXAA00706
qfRAA27175 xfAAA00784 xfDAA01360 xfFAA01616

文件名字总是由一个两字符的前缀加上一个随机数字。前缀有四钟:
df:邮件内容
qf:邮件头和一些控制信息
xf:一些临时文件
tf:qf 文件的临时存储文件

可以通过看队列中的 qf 文件来确定当前队列信息，不过通常都可以用 mailq 程序来完成 对队列状态的查询:
[root@mail mqueue]# mailq
Mail Queue (1 request)
--Q-ID-- --Size-- -----Q-Time----- ------------Sender/Recipient---------- -
RAA27175 22429 Fri Feb 25 17:34
: deferred)
chaszhj@21cn.net

这表示队列中目前只有一封信，由 zhangfl 发出，收信人是 chaszhj@21cn.net。

由 sendmail 接受到的信件在用户取走之前将暂时存储在/var/spool/mail 目录下面，存 储的方法非常简单，就是每个有待读邮件的用户一个文件:

[root@mail spool]# ls -l /var/spool/mail |more
total 19364
-rw-rw---- 1 anyi mail 7559035 Feb 28 22:04 anyi
-rw-rw---- 1 baixuan mail 514 Nov 7 01:32 baixuan
-rw-rw---- 1 cwc mail 515 Feb 28 08:35 cwc

如果一个用户有多封电子邮件，那么这些邮件就被简单地连接在一起构成一个大文件 (所以你会看到非常巨大的邮件文件)。

如果你面对的是一个很庞大的邮件服务系统，有时会在某个邮件队列中拥塞太多的消 息，你可能想把这个队列暂停并且在网络空闲的时候再发送，暂停一个邮件队列的办法非常简单，就是将/var/spool/mqueue 目录移走:
killall sendmail
mv /var/spool/mqueue /var/spool/mqueue.stop
mkdir /var/spool/mqueue
sendmail -bd

当网络空闲的时候，可以用-oQ 参数立即处理拥塞的队列:
sendmail -oQ /var/spool/mqueue.stop -q

649 ☆SMTP 协议的基本命令

在/etc/mail/helpfile 中有描述
为了使用这种测试技术，你需要了解 SMTP 协议的基本命令，这样的命令有 14 个，在下面列出:

HELO
标志发起 smtp 请求的主机，例如，从 client1 发起 smtp 会话，可以使用
HELO client1

MAIL FROM:
启动一个邮件会话，在这个行中需要标志发信人的信封地址，例如，要从 user1@client1 发出邮件，使用 MAIL FROM:user1@client1，注意尖括号的用法。

RCPT TO:
标志收信人的信封地址，例如，要发送给 user2@mail2，使用 RCPT TO: user2@mail2。在一个 MAIL FROM 之后可以给出多个收信人地址，以便实现多副本的传送。

VRFY

验证某个地址，例如，要确定 test@mail 是一个可以投递的地址，使用 VRFY test@mail。

EXPN
显示某个收件人地址或者用户名的实际名字。例如，要显示 postmaster 用户的实际投 递地址，使用 EXPN postmaster。如果在某个用户的目录下有.forward 文件，这个文件的 内容将会被自动使用。

DATA
开始写信，在 MAIL 和 RCPT 之后可以使用这个命令传输信件正文，传输完毕之后输入一 个.退出。

QUIT
关闭 smtp 会话

RSET
复位连接状态

HELP
显示这个命令表。

650 ☆sendmail 测试

/etc/mail 目录下配置一些文档。
cd /etc/mail
echo 'examply.com' >> local-host-names 接收邮件的主机名
echo 'localhost RELAY' >> access 用来拒绝或允许来自某个域的邮件，本例允许本地转发。
makemap hash access < access 生成 access.db 数据库
touch domaintable 用来把旧域名映射互新域名
makemap hash domaintable < domaintable
touch mailertable 来覆盖向指定域的路由
makemap hash mailertable < mailertable
touch trusted-users
touch virtusertable 用来把用户和域名映射到其它地址
makemap hash virtusertable < virtusertable
chown root:wheel /var/spool/mqueue/
chmod 700 /var/spool/mqueue
touch aliases 别名数据库，文本形式。可参照源码目录树下 sendmail/aliases 文件。
newaliases 从文本文件中创建一个新的别名数据库文件。
sendmail -v -bi 调试启动。
/etc/mail/aliases: 42 aliases, longest 10 bytes, 432 bytes total
如果出现以上提示信息，则启动成功。可用以下命令正式启动:

打开你喜欢的 Mail Client，设置 smtp 服务器为:localhost (假如和服务器在同一台机子上，也可以设置本机的 IP),端口为:25.然后随便一段测试信息,填写好接收邮件的地址，按发送。你的 Mail Client 会提示邮件已发送。实际上你的邮件还在计算机上排队，并未发出！要发送所有排队的邮件，你需要连上网，然后你 root 身份登陆，并运行:# sendmail -q .现在你的邮件就会发送到指定的邮箱中。

651 ☆postfix 安装

需要:postfix

652 ☆postfix 收取邮件

为了安全的原因，sendmail 和 postfix 的缺省的配置允许发邮件但是不允许从网络上接收邮件(缺省的它们只接受从回环接口上的连接)。
修改/etc/postfix/main.cf

找到并注释如下行
inet_interfaces = localhost

取消注释该行:
inet_interfaces = all

653 ☆Postfix 启动和校验

运行
service sendmail stop

使用图形工具
system-switch-mail
使得 postfix 成为活跃的 MTA。

也可以使用如下的命令行:
alternatives -set mta /usr/sbin/sendmail.postfix

启动命令如下:
service postfix restart

确定 hostname 命令正确的返回您的主机名称。确保 DNS 配置正确应该是您的 FQDN。

如果 sendmail 返回您的主机名称为 localhost,您可能错误配置了/etc/hosts 文件。检查您的/etc/hosts 文件，删除所有的但记住留下 localhost 的指向，然后再试一遍。如果/etc/hosts 文件是正确的，那

么检查一下在/etc/sysconfig/netwoek 中的 HOSTNAME 的定义。当这些值都正确的时候，启动 postfix 服务。

确定 postfix 在启动的时候没有错误
Red Hat Linux 的安装使用提供的 syslog 工具来记录所有的信息到文件/var/log/maillog 中去。检查此文件中的最后查找任何错误信息。

试图向 root@server1 发送简单的邮件并且检查/var/log/maillog 的记录文件
mail -s `echo $USER` root@panda < /etc/redhat-release

应该如下所示:
Jul 26 20:55:10 localhost postfix/pickup[4902]: 717AB335FE: uid=0 from=<root>
Jul 26 20:55:10 localhost postfix/cleanup[4943]: 717AB335FE:
message-id=<20060726125510.717AB335FE@panda.com>
Jul 26 20:55:10 localhost postfix/qmgr[4903]: 717AB335FE: from=<root@panda.com>, size=314, nrcpt=1 (queue active)
Jul 26 20:55:10 localhost postfix/local[4945]: 717AB335FE: to=<root@panda.com>, relay=local, delay=0, status=sent (delivered to mailbox)
Jul 26 20:55:10 localhost postfix/qmgr[4903]: 717AB335FE: removed

654 ☆Postfix 的别名

在 postfix 决定消息的接受者的目的地的之前，其先试图在别名中查找。
postfix 的主要的别名配置文件是/etc/aliases。为了优化查找，postfix 为其别名记录建立了一个哈希表别名数据库/etc/aliases.db(和 sendmail 类似).该文件通过 newaliases 命令产生。

下列命令将增加用户 student(如果不存在的话)
useradd student

在/etc/aliases 行加入如下的行:
注意:注释 root 别名的那一行为 postfix
me: student
wizards: root, me
methere: panda@panda.com

现在运行
newaliases
更新数据库

尝试发送邮件给您定义的收件人:
echo "hello there" | mail -s "hello" me
echo "hello there" | mail -s "hello" wizards

echo "hello there" | mail -s "hello" methere

是否所有的位于 wizards 的收件人都受到了邮件？
要确认/etc/mail/access 中打开权限

655 ☆postfix 允许转发

缺省的 postfix 允许在子网上的任何人通过您的机器进行转发。但是并不是在每一个环境中都安全的。例如，您的机器和其他机器在一起，如果您的本地子网里有一台机器被其他人控制，那么其他的机器都会有麻烦。

让您的伙伴扮演恶意的垃圾邮件的发送者，该人能够通过 telnet 到您的机器上的 postfix 的 25 号断口，进行垃圾邮件发送地址的欺骗，在 panda 键入如下命令:

[root@panda mail]# telnet panda 25
Trying 127.0.0.1...
Connected to panda.com (127.0.0.1).
Escape character is '^]'.
220 panda.com ESMTP Postfix

helo root.panda.com
250 panda.com

mail from:panda@panda.com
250 Ok

rcpt to: root@panda.com
250 Ok

data
354 End data with <CR><LF>.<CR><LF>

Subject: Faked
this was faked!
.
250 Ok: queued as 9DB2733601

quit
221 Bye
Connection closed by foreign host.

垃圾邮件现在送到您的机器上了。下一步，看看您的伙伴能不能从您的机器转发给第三台机器:

由于您的机器已经被配置成为允许混杂转发，垃圾邮件可以通过您的机器进行邮件转发。

注意/var/log/maillog 的变化

656 ☆postfix 不允许转发

编辑文件/etc/postfix/main.cf 取消转发。
查找并且取消注释下面的行，
mynetworks_style = host

并且重新启动 postfix

让您的伙伴再从 stationY 转发垃圾邮件。您的 postfix 还是一个转发器么？任何一个转发的都会产生如下的消息:

554 <root@panda.panda.com>: Recipient address rejected: Relay access denied

657 ☆postfix 选择性的转发

对于特定的主机，域或者网络，编辑/etc/postfix/main.cf 并且重新启动 postfix。对于特定的主机允许通过您的机器进行转发，找到并且取消注释该行:
mynetworks_style = host

然后添加新行来允许转发的主机和网络，在这里允许 station1 和本地转发
mynetworks = 192.168.152.128, 127.0.0.0/8

658 ☆postfix 接收和转发的条件

(1)默认情况下，postfix 接收符合以下条件的邮件：
目的地为$inet interfaces 的邮件；
目的地为$mydestination 的邮件；
目的地为$virtual_alias_maps 的邮件。
(2)默认情况下，postfix 转发符合以下条件的邮件：
来自客户端 IP 地址符合$mynetworks 的邮件；
来自客户端主机名称符合 relay_domains 及其子域的邮件；
目的地为$relay_domains 及其子域的邮件。

659 ☆main.cf

修改/etc/postfix/main.cf 的配置:

myhostname = mail.cngnu.org
指定运行 postfix 服务的邮件主机的主机名称(FQDN 名)

mydomain = cngnu.org
指定运行 postfix 服务的邮件主机的域名称

myorigin = $mydomain
设置由本台邮件主机寄出的每封邮件的邮件头中 mail from 的地址

inet interfaces = all
默认情况下，inet interfaces 参数的值被设置为 localhost，这表明只能在本地邮件主机上寄信。如果邮件主机上有多个网络接口，而又不想使全部的网络接口都开放 Postfix 服务，就可以用主机名指定需要开放的网络接口。不过，通常是将所有的网络接口都开放，以便接收从任何网络接口来的邮件，即将 inet interfaces 参数的值设置为“all”。

mydestination = $mydomain,$myhostname
只有当发来的邮件的收件人地址与该参数值相匹配时，Postfix 才会将该邮件接收下来。例如，这里将该参数值设置为$mydomain,$myhosname，表明无论来信的收件人地址是 xxx@cngnu.org(其中，XXX 表示某用户的邮件账户名)，还是 xxx@mail.cngnu.org，Postfix 都会接收这些邮件。

mynetworks_style = host

mynetworks = 127.0.0.0/8,192.168.1.0/24
设置可转发(Relay)哪些网络的邮件.可以使用 mynetworks 参数来设置。可将该参数值设置为所信任的某台主机的 IP 地址，也可设置为所信任的某个 IP 子网或多个 IP 子网(用“，”或者“”分隔)。这里，将 mynetworks 参数值设置为 192.168.16.0／24，则表示这台邮件主机只转发子网 192.168.16.0／24 中的客户端所发来的邮件，而拒绝为其他子网转发邮件。

relay_domains = gdvcp.net
mynetworks 参数是针对邮件来源的 IP 来设置的，而 relay_domains 参数则是针对邮件来源的域名或主机名来设置的。例如，将该参数值设置为 gdvcp.net，则表示任何由域 gdvcp.net 发来的邮件都会被认为是信任的，Postfix 会自动对这些邮件进行转发。

virtual_alias_domains = dzxx.cn,panda.com
用来指定虚拟别名域的名称

virtual_maps = hash:/etc/postfix/virtual　　#,mysql:/etc/postfix/virtual.mysql
含有虚拟别名域定义的文件路径。

alias_maps = hash:/etc/aliases
指定含有用户别名定义的文件路径

alias_database = hash:/etc/aliases
指定别名表数据库文件路径。

home_mailbox = Maildir/
mailbox_transport = cyrus
fallback_transport = cyrus

smtpd_sasl_auth_enable = yes
指定是否要启用 SASL 作为 SMTP 认证方式。默认不启用，这里必须将它启用，所以要将该参数值设置为“yes”。
smtpd_sasl_local_domain = ''
如果采用 Cyrus-SASL V2 版进行认证，那么这里不作设置。

smtpd_recipient_restrictions = permit_mynetworks,permit_sasl_authenticated,reject_unauth_destination
表示通过收件人地址对客户端发来的邮件进行过滤。通常有以下几种限制规则。
permit_mynetworks:表示只要是收件人地址位于 mynetworks 参数中指定的网段就可以被转发邮件
permit_sasl_authenticated:表示允许转发通过 SASL 认证的邮件。
reject_unauth_destination:表示拒绝转发含未信任的目标地址的邮件。

broken_sasl_auth_clients = yes
表示是否兼容非标准的 SMTP 认证。有一些 Microsoft 的 SMTP 客户端(如 Outlook Express 4.x)采用非标准的 SMTP 认证协议，只需将该参数设置为"yes"就可解决这类不兼容问题。

smtpd_client_restrictions = permit_sasl_authenticated
表示限制可以向 postfix 发起 SMTP 连接的客户端。如果要禁止未经过认证的客户端向 postfix 发起 SMTP 连接，则可将该参数值设置为 permit_sasl_authenticated

smtpd_sasl_security_options = noanonymous
用来限制某些登录的方式。如果将该参数值设置为 noanonymous，则表示禁止采用匿名登录方式。

```
    permit_mynetworks,
    permit_sasl_authenticated,
    check_recipient_access mysql:/etc/postfix/filter.mysql,
    reject_invalid_hostname,
    reject_non_fqdn_hostname,
    reject_unknown_sender_domain,
    reject_non_fqdn_sender,
    reject_non_fqdn_recipient,
    reject_unknown_recipient_domain,
    reject_unauth_pipelining,
    reject_unauth_destination,
    permit
```

如果希望支持更多的虚拟域，可以在 mydestination 参数后面加上你所要支持的域即可。

通过 virtual 和 virtual.mysql 为系统提供了邮箱本地查询表。

660☆virtual

/etc/postfix/virtual
虚拟别名　　实际名
@dzxx.cn　　@gdvcp.net
admin@panda.com　　lbt
st123@panda.com　　st123001,st123002,st123003
daliu@panda.comlbt,liu6812@163.com

661☆master.cf

由 postfix 运行的守护程序

662☆pcre_table

地址改写的或邮件路由的 perl 表达式

663☆postfix-files

postfix 文件和权限

664☆postfix-script

管理 postfix,类似 start,stop

665☆regexp_table

从 postfix 查找账号的表
用 postconf -m 来检查

666☆relocated

用户转移到新位置,用这里来重定向

667☆transport

指定一个传输协议

668 ☆postconf

postconf
测试配置文件

postconf -n
看当前配置文件的路径

669 ☆重启 postfix

修改 main.cf 和 virtual,aliases 后

postmap /etc/postfix/virtual
生成 etc/postfix/virtual.db

postalias /etc/aliases
生成/etc/aliases.db

postfix reload
重载 main.cf

670 ☆SMTP 认证的配置

如果任何人都可以通过一台邮件服务器来转发邮件，会有什么后果呢?很可能这台邮件服务器就成为了各类广告与垃圾信件的集结地或中转站，网络带宽也会很快被耗尽。为了避免这种情况的出现，MTA 默认不会对外开放转发功能，而仅对本机(1ocalhost)开放转发功能。但是，在实际应用中，必须在 MTA 主配置文件中通过设置 mynetworks、relay domains 参数来开放一些所信任的网段或网域，否则该邮件服务器几乎没有什么用途。在开放了这些所信任的网段或网域后，还可以通过设置 SMTP 认证，对要求转发邮件的客户端进行用户身份(用户账户名与密码)验证。只有通过了验证，才能接收该用户寄来的邮件并帮助转发。

目前，比较常用的 SMTP 认证机制是通过 Cyrus-SASI.。包来实现的。

671 ☆Cyrus-SASL

是 Cyrus Simple Authentication and Security Layer 的简写，它最大的功能是为应用程序提供了认证函数库。应用程序可以通过函数库所提供的功能定义认证方式，并让 SASL。通过与邮件服务器主机的沟通从而提供认证的功能。

672 ☆Cyrus-SASL 安装

rpm -qa | grep sasl

673 ☆Cyrus-SASL 启动

运行 saslauthd 守护进程
service saslauthd restart

674 ☆Cyrus-SASL 认证机制

默认情况下，Cyrus-SASL V2 版使用 saslauthd 这个守护进程进行密码认证，而密码认证的方法有多种，使用下面的命令可查看当前系统中的 Cyrus-SASL V2 所支持的密码验证机制。
[root@panda mail]# saslauthd -v
saslauthd 2.1.19
authentication mechanisms: getpwent kerberos5 pam rimap shadow ldap

675 ☆saslauthd 配置

配置 saslauthd 使用 shadow 认证方案:
/etc/sysconfig/saslauthd
MECH=shadow

配置 saslauthd 使用 PAM 认证方案(非默认,需要配置 pam,否则测试失败):
/etc/sysconfig/saslauthd
MECH=pam

命令行方式启动验证方案:
saslauthd -a shadow
用 shadow 的用户和密码进行验证

676 ☆测试 Cyrus-SASL

ps aux | grep saslauthd

service saslauthd restart

testsaslauthd -u userid -p password

[root@panda ~]# testsaslauthd -u student -p '111111'
0: OK "Success."
如果出现以上信息，就说明 saslauthd 正常运行了。表示认证功能有效

服务器启动后，可以用 telnet 连接服务器。
telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 test.tigerhead ESMTP Sendmail 8.12.10/8.12.10; Tue, 30 Mar 2004 14:50:14 +0800
ehlo test (!!!!!!!!注意命令是 ehlo)你输入的命令，按回车结束。
250-test.tigerhead Hello LOCALHOST.localdomain [127.0.0.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN　　#就表明 SMTP 认证成功
250-DELIVERBY
250 HELP
以 250-开头的为服务器的响应信息。注意倒数第三行，这就是成功配置 smtp 验证的显示。
输入 quit 离开。

677 ☆Cyrus-SASL 配置 MTA

postfix 中
更改 main.cf 中的配置
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain =
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes

建立/usr/lib/sasl2/smtpd.conf
设置 Postfix 使用 SASL 的 saslauthd 认证守护进程来支持 smtp auth 认证，并只打开了 plain 和 login 认证模块:
[root@mail root]# echo pwcheck_method:saslauthd > /usr/lib/sasl2/smtpd.conf
[root@mail root]# echo mech_list: plain login >> /usr/lib/sasl2/smtpd.conf

sendmail 中
更改/etc/mail/sendmail.mc
去掉下面俩行的注释
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl

define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
如果 sendmail 没办法收信,就把
DAEMON_OPTIONS(`Port=587, Name=MSA, M=Ea')dnl 改为
DAEMON_OPTIONS(`Port=25, Name=MSA')dnl (这一句不能和 DAEMON_OPTIONS(`Port=25, Name=MTA')dnl
同时存在)

建立/usr/lib/sasl2/Sendmail.conf
当 sendmail 要使用 SMTP 认证时，必须创建一个 SASL 的配置文件来把 MTA 程序定义成一个 SASL 应用。配置文件名为 Sendmail.conf(注意是大写的 S),位于/usr/lib/sasl2 目录中，也就是 /usr/local/sasl2/lib/sasl2 这个目录，记得上面新建的链接了吗？在该文件中你定义你希望使用的认证数据库方法，以下这个例子使用 saslauthd 来验证认证请求。
echo 'pwcheck_method: saslauthd' > /usr/lib/sasl2/Sendmail.conf(注意大小写)

如果是 pwcheck_method: pam 的话是直接调用 pam 认证
如果是 pwcheck_method: saslauth 的话要启用 saslauth -a pam ，让 saslauth 调用 pam 认证

将这个文件复制成 smtpd.conf，即 cp Sendmail.conf smtpd.conf，这个文件也要在/usr/lib/sasl2/目录下

在/etc/pam.d/目录下参照其它文件建立一个“smtp”文件(postfix 的是 smtp.postfix)，内容如下：
#%PAM-1.0
auth required pam_stack.so service=system-auth
account required pam_stack.so service=system-auth

678 ☆dovecot 配置

postfix 服务只是一个 MTA(邮件传输代理)，它只提供 SMTP 服务，也就是只提供邮件的转发及本地的分发功能。要实现邮件的异地接收，还必须安装 POP 或 IMAP 服务。

dovecot 提供了这些服务:POP3,POP3S,IMAP,IMAPS

编辑/etc/dovecot.conf 文件

在 protocols=一行，加入你需要的的服务
缺省的 ports:
imap: 143
imaps: 993
pop3: 110
pop3s: 995

ssl_cert_file 和 ssl_key_file,可以看到dovecot是拿什么文件来做ssl认证的，可以自己创建新的pem文件。

将 disable_plaintext_auth 配置为 no(非缺省)

将 auth_mechanisms 配置为 plain(缺省的)

将 auth_userdb 配置为 passwd
也就是说缺省为所有 local user 都能登录。

将 auth_passdb 设置为 shadow
或者将 auth_passdb 设置为 pam

增加对 pop3 配置文件。（缺省没有，要自己建立）
/etc/pam.d/pop3 文件内容如：
代码:

```
auth        required      /lib/security/pam_stack.so service=system-auth
auth        required      pam_unix.so
auth        required      pam_listfile.so item=user sense=deny file=/etc/security/dovecot.deny
onerr=fail
account     required      /lib/security/pam_stack.so service=system-auth
#account                 required      pam_access.so
account     required      pam_unix.so
```

用了 pam_listfile.so 模块，当然也可以用 pam_access.so 模块来限定。

679☆dovecot.conf

```
## Dovecot 1.0 configuration file

# Default values are shown after each value, it's not required to uncomment
# any of the lines. Exception to this are paths, they're just examples
# with real defaults being based on configure options. The paths listed here
# are for configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var
# --with-ssldir=/usr/share/ssl

# 运行时存储数据的目录
#base_dir = /var/run/dovecot/

# 加入你需要的的服务:
#   imap imaps pop3 pop3s
```

```
#protocols = imap imaps

#  监听的 IP 或者主机地址.
# It's not currently possible to specify multiple addresses.
# "*"  是指定监听所有 IPv4 的 interfaces.
# "[::]"  监听所有 IPv6 的 interfaces,可能会监听所有的 IPv4 的 interfaces 但是这个取决于操作系统
#  指定端口的格式是:"host:port".
imap_listen = [::]
pop3_listen = [::]

#  监听 SSL 连接的 IP 或主机地址.
# Defaults to above non-SSL equilevants if not specified.
#imaps_listen =
#pop3s_listen =

#  禁用 SSL/TLS 支持.
#ssl_disable = no

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self-signed
# certificate,  要保证在 dovecot-openssl.cnf 中更新域
#ssl_cert_file = /usr/share/ssl/certs/dovecot.pem
#ssl_key_file = /usr/share/ssl/private/dovecot.pem

# SSL parameter file. Master process generates this file for login processes.
# It contains Diffie Hellman and RSA parameters.
#ssl_parameters_file = /var/run/dovecot/ssl-parameters.dat

# How often to regenerate the SSL parameters file. Generation is quite CPU
# intensive operation. The value is in hours, 0 disables regeneration
# entirely.
#ssl_parameters_regenerate = 24

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that 127.*.*.* and
# IPv6 ::1 addresses are considered secure, this setting has no effect if
# you connect from those addresses.
#disable_plaintext_auth = yes

# Use this logfile instead of syslog(). /dev/stderr can be used if you want to
# use stderr for logging (ONLY /dev/stderr - otherwise it is closed).
```

```
#log_path =

# For informational messages, use this logfile instead of the default
#info_log_path =

# Prefix for each line written to log file. % codes are in strftime(3)
# format.
#log_timestamp = "%b %d %H:%M:%S "

##
## Login processes
##

# Directory where authentication process places authentication UNIX sockets
# which login needs to be able to connect to. The sockets are created when
# running as root, so you don't have to worry about permissions. Note that
# everything in this directory is deleted when Dovecot is started.
login_dir = /var/run/dovecot-login

# chroot login process to the login_dir. Only reason not to do this is if you
# wish to run the whole Dovecot without roots.
#login_chroot = yes


##
## IMAP login process
##

login = imap

# Executable location.
#login_executable = /usr/libexec/dovecot/imap-login

# User to use for the login process. Create a completely new user for this,
# and don't use it anywhere else. The user must also belong to a group where
# only it has access, it's used to control access for authentication process.
#login_user = dovecot

# Set max. process size in megabytes. If you don't use
# login_process_per_connection you might need to grow this.
#login_process_size = 32
```

```
# Should each login be processed in it's own process (yes), or should one
# login process be allowed to process multiple connections (no)? Yes is more
# secure, espcially with SSL/TLS enabled. No is faster since there's no need
# to create processes all the time.
#login_process_per_connection = yes

# Number of login processes to create. If login_process_per_user is
# yes, this is the number of extra processes waiting for users to log in.
#login_processes_count = 3

# Maximum number of extra login processes to create. The extra process count
# usually stays at login_processes_count, but when multiple users start logging
# in at the same time more extra processes are created. To prevent fork-bombing
# we check only once in a second if new processes should be created - if all
# of them are used at the time, we double their amount until limit set by this
# setting is reached. This setting is used only if login_process_per_use is yes.
#login_max_processes_count = 128

# Maximum number of connections allowed in login state. When this limit is
# reached, the oldest connections are dropped. If login_process_per_user
# is no, this is a per-process value, so the absolute maximum number of users
# logging in actually login_processes_count * max_logging_users.
#login_max_logging_users = 256

##
## POP3 login process
##

# Settings default to same as above, so you don't have to set anything
# unless you want to override them.

login = pop3

# Exception to above rule being the executable location.
#login_executable = /usr/libexec/dovecot/pop3-login

##
## Mail processes
##

# Maximum number of running mail processes. When this limit is reached,
# new users aren't allowed to log in.
```

```
#max_mail_processes = 1024

# Show more verbose process titles (in ps). Currently shows user name and
# IP address. Useful for seeing who are actually using the IMAP processes
# (eg. shared mailboxes or if same uid is used for multiple accounts).
#verbose_proctitle = no

# Show protocol level SSL errors.
#verbose_ssl = no

# Valid UID range for users, defaults to 500 and above. This is mostly
# to make sure that users can't log in as daemons or other system users.
# Note that denying root logins is hardcoded to dovecot binary and can't
# be done even if first_valid_uid is set to 0.
#first_valid_uid = 500
#last_valid_uid = 0

# Valid GID range for users, defaults to non-root/wheel. Users having
# non-valid GID as primary group ID aren't allowed to log in. If user
# belongs to supplementary groups with non-valid GIDs, those groups are
# not set.
#first_valid_gid = 1
#last_valid_gid = 0

# Grant access to these extra groups for mail processes. Typical use would be
# to give "mail" group write access to /var/mail to be able to create dotlocks.
#mail_extra_groups =

# ':' separated list of directories under which chrooting is allowed for mail
# processes (ie. /var/mail will allow chrooting to /var/mail/foo/bar too).
# This setting doesn't affect login_chroot or auth_chroot variables.
# WARNING: Never add directories here which local users can modify, that
# may lead to root exploit. Usually this should be done only if you don't
# allow shell access for users. See doc/configuration.txt for more information.
#valid_chroot_dirs =
# Default chroot directory for mail processes. This can be overridden by
# giving /./ in user's home directory (eg. /home/./user chroots into /home).
#mail_chroot =

# Default MAIL environment to use when it's not set. By leaving this empty
# dovecot tries to do some automatic detection as described in
# doc/mail-storages.txt. There's a few special variables you can use:
```

```
#
#    %u - username
#    %n - user part in user@domain, same as %u if there's no domain
#    %d - domain part in user@domain, empty if user there's no domain
#    %h - home directory
#
# You can also limit a width of string by giving the number of max. characters
# after the '%' character. For example %1u gives the first character of
# username. Some examples:
#
#    default_mail_env = maildir:/var/mail/%1u/%u/Maildir
#    default_mail_env = mbox:~/mail/:INBOX=/var/mail/%u
#    default_mail_env = mbox:/var/mail/%d/%n/:INDEX=/var/indexes/%d/%n
#
#default_mail_env =

# Space-separated list of fields to cache for all mails. Currently these
# fields are allowed followed by a list of commands they speed up:
#
#   Envelope          - FETCH ENVELOPE and SEARCH FROM, TO, CC, BCC, SUBJECT,
#                       SENTBEFORE, SENTON, SENTSINCE, HEADER MESSAGE-ID,
#                       HEADER IN-REPLY-TO
#   Body               - FETCH BODY
#   Bodystructure - FETCH BODY, BODYSTRUCTURE
#   MessagePart     - FETCH BODY[1.2.3] (ie. body parts), RFC822.SIZE,
#                       SEARCH SMALLER, LARGER, also speeds up BODY/BODYSTRUCTURE
#                       generation. This is always set with mbox mailboxes, and
#                       also default with Maildir.
#
# Different IMAP clients work in different ways, that's why Dovecot by default
# only caches MessagePart which speeds up most operations. Whenever client
# does something where caching could be used, the field is automatically marked
# to be cached later. For example after FETCH BODY the BODY will be cached
# for all new messages. Normally you should leave this alone, unless you know
# what most of your IMAP clients are. Caching more fields than needed makes
# the index files larger and generate useless I/O.
#
# With maildir there's one extra optimization - if nothing is cached, indexing
# the maildir becomes much faster since it's not opening any of the mail files.
# This could be useful if your IMAP clients access only new mails.

#mail_cache_fields = MessagePart
```

```
# Space-separated list of fields that Dovecot should never set to be cached.
# Useful if you want to save disk space at the cost of more I/O when the fields
# needed.
#mail_never_cache_fields =

# Workarounds for various client bugs:
#   oe6-fetch-no-newmail:
#     Never send EXISTS/RECENT when replying to FETCH command. Outlook Express
#     seems to think they are FETCH replies and gives user "Message no longer
#     in server" error. Note that OE6 still breaks even with this workaround
#     if synchronization is set to "Headers Only".
#   outlook-idle:
#     Outlook and Outlook Express never abort IDLE command, so if no mail
#     arrives in half a hour, Dovecot closes the connection. This is still
#     fine, except Outlook doesn't connect back so you don't see if new mail
#     arrives.
#   outlook-pop3-no-nuls:
#     Outlook and Outlook Express hang if mails contain NUL characters.
#     This setting replaces them with 0x80 character.
#client_workarounds =

# Dovecot can notify client of new mail in selected mailbox soon after it's
# received. This setting specifies the minimum interval in seconds between
# new mail notifications to client - internally they may be checked more or
# less often. Setting this to 0 disables the checking.
# NOTE: Evolution client breaks with this option when it's trying to APPEND.
#mailbox_check_interval = 0

# Like mailbox_check_interval, but used for IDLE command.
#mailbox_idle_check_interval = 30

# Allow full filesystem access to clients. There's no access checks other than
# what the operating system does for the active UID/GID. It works with both
# maildir and mboxes, allowing you to prefix mailboxes names with eg. /path/
# or ~user/.
#mail_full_filesystem_access = no

# Maximum allowed length for custom flag name. It's only forced when trying
# to create new flags.
#mail_max_flag_length = 50
```

```
# Save mails with CR+LF instead of plain LF. This makes sending those mails
# take less CPU, especially with sendfile() syscall with Linux and FreeBSD.
# But it also creates a bit more disk I/O which may just make it slower.
#mail_save_crlf = no

# Use mmap() instead of read() to read mail files. read() seems to be a bit
# faster with my Linux/x86 and it's better with NFS, so that's the default.
#mail_read_mmaped = no

# By default LIST command returns all entries in maildir beginning with dot.
# Enabling this option makes Dovecot return only entries which are directories.
# This is done by stat()ing each entry, so it causes more disk I/O.
# (For systems setting struct dirent->d_type, this check is free and it's
# done always regardless of this setting)
#maildir_stat_dirs = no

# Copy mail to another folders using hard links. This is much faster than
# actually copying the file. This is problematic only if something modifies
# the mail in one folder but doesn't want it modified in the others. I don't
# know any MUA which would modify mail files directly. IMAP protocol also
# requires that the mails don't change, so it would be problematic in any case.
# If you care about performance, enable it.
#maildir_copy_with_hardlinks = no

# Check if mails' content has been changed by external programs. This slows
# down things as extra stat() needs to be called for each file. If changes are
# noticed, the message is treated as a new message, since IMAP protocol
# specifies that existing messages are immutable.
#maildir_check_content_changes = no

# Which locking methods to use for locking mbox. There's three available:
#   dotlock: Create <mailbox>.lock file. This is the oldest and most NFS-safe
#            solution. If you want to use /var/mail/ like directory, the users
#            will need write access to that directory.
#   fcntl  : Use this if possible. Works with NFS too if lockd is used.
#   flock  : May not exist in all systems. Doesn't work with NFS.
#
# You can use both fcntl and flock too; if you do the order they're declared
# with is important to avoid deadlocks if other MTAs/MUAs are using both fcntl
# and flock. Some operating systems don't allow using both of them
# simultaneously, eg. BSDs. If dotlock is used, it's always created first.
mbox_locks = fcntl
```

```
# Should we create dotlock file even when we want only a read-lock? Setting
# this to yes hurts the performance when the mailbox is accessed simultaneously
# by multiple processes, but it's needed for reliable reading if no other
# locking methods are available.
#mbox_read_dotlock = no

# Maximum time in seconds to wait for lock (all of them) before aborting.
#mbox_lock_timeout = 300

# If dotlock exists but the mailbox isn't modified in any way, override the
# lock file after this many seconds.
#mbox_dotlock_change_timeout = 30

# umask to use for mail files and directories
#umask = 0077

# Drop all privileges before exec()ing the mail process. This is mostly
# meant for debugging, otherwise you don't get core dumps. Note that setting
# this to yes means that log file is opened as the logged in user, which
# might not work. It could also be a small security risk if you use single UID
# for multiple users, as the users could ptrace() each others processes then.
#mail_drop_priv_before_exec = no

##
## IMAP process
##

# Executable location
#imap_executable = /usr/libexec/dovecot/imap

# Set max. process size in megabytes. Most of the memory goes to mmap()ing
# files, so it shouldn't harm much even if this limit is set pretty high.
#imap_process_size = 256

# Support for dynamically loadable modules.
#imap_use_modules = no
#imap_modules = /usr/lib/dovecot/imap

##
## POP3 process
##
```

```
# Executable location
#pop3_executable = /usr/libexec/dovecot/pop3

# Set max. process size in megabytes. Most of the memory goes to mmap()ing
# files, so it shouldn't harm much even if this limit is set pretty high.
#pop3_process_size = 256

# Support for dynamically loadable modules.
#pop3_use_modules = no
#pop3_modules = /usr/lib/dovecot/pop3

##
## Authentication processes
##

# An Authentication process is a child process used by Dovecot that
# handles the authentication steps. The steps cover an authentication
# mechanism (auth_mechanisms, how the client authenticates in the IMAP or
# POP3 protocol), which password database should be queried (auth_passdb),
# and which user database should be queried (auth_userdb, to obtain
# UID, GID, and location of the user's mailbox/home directory).
#
# You can have multiple processes, though a typical configuration will
# have only one. Each time "auth = xx" is seen, a new process
# definition is started. The point of multiple processes is to be able
# to set stricter permissions. (See auth_user below.)
#
# Just remember that only one Authentication process is asked for the
# password, so you can't have different passwords accessible through
# different process definitions (unless they have different
# auth_mechanisms, and you're ok with having different password for
# each mechanisms).

# Authentication process name.
auth = default

# Specifies how the client authenticates in the IMAP protocol.
# Space separated list of permitted authentication mechanisms:
#      anonymous plain digest-md5 cram-md5
#
# anonymous - No authentication required.
```

```
# plain - The password is sent as plain text. All IMAP/POP3 clients
#    support this, and the password can be encrypted by Dovecot to match
#    any of the encryption schemes used in password databases.
# digest-md5 and cram-md5 - both encrypt the password so it is more
#    secure in transit, but are not well supported by clients, and
#    require that the password database use a matching encryption
#    scheme (or be in plaintext).
#
# See auth.txt for more details.
#
# If you are using SSL there is less benefit to digest-md5 and
# cram-md5 as the communication is already encrypted.
auth_mechanisms = plain

# Space separated list of realms for SASL authentication mechanisms that need
# them. You can leave it empty if you don't want to support multiple realms.
# Many clients simply use the first one listed here, so keep the default realm
# first.
#auth_realms =

# Default realm/domain to use if none was specified. This is used for both
# SASL realms and appending @domain to username in plaintext logins.
#auth_default_realm =

# Where user database is kept:
#      passwd: /etc/passwd or similiar, using getpwnam()
#      passwd-file <path>: passwd-like file with specified location
#      static uid=<uid> gid=<gid> home=<dir template>: static settings
#      vpopmail: vpopmail library
#      ldap <config path>: LDAP, see doc/dovecot-ldap.conf
#      pgsql <config path>: a PostgreSQL database, see doc/dovecot-pgsql.conf
auth_userdb = passwd

# Where password database is kept:
#      passwd: /etc/passwd or similiar, using getpwnam()
#      shadow: /etc/shadow or similiar, using getspnam()
#      pam [<service> | *]: PAM authentication
#      passwd-file <path>: passwd-like file with specified location
#      vpopmail: vpopmail authentication
#      ldap <config path>: LDAP, see doc/dovecot-ldap.conf
#      pgsql <config path>: a PostgreSQL database, see doc/dovecot-pgsql.conf
#auth_passdb = pgsql /usr/local/etc/dovecot-pgsql.conf
```

```
auth_passdb = pam

#auth_executable = /usr/libexec/dovecot/dovecot-auth

# Set max. process size in megabytes.
#auth_process_size = 256

# User to use for the process. This user needs access to only user and
# password databases, nothing else. Only shadow and pam authentication
# requires roots, so use something else if possible. Note that passwd
# authentication with BSDs internally accesses shadow files, which also
# requires roots.
auth_user = root

# Directory where to chroot the process. Most authentication backends don't
# work if this is set, and there's no point chrooting if auth_user is root.
#auth_chroot =

# Number of authentication processes to create
#auth_count = 1

# List of allowed characters in username. If the user-given username contains
# a character not listed in here, the login automatically fails. This is just
# an extra check to make sure user can't exploit any potential quote escaping
# vulnerabilities with SQL/LDAP databases. If you want to allow all characters,
# set this value to empty.
#auth_username_chars =
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890.-_@

# Username to use for users logging in with ANONYMOUS SASL mechanism
#auth_anonymous_username = anonymous

# More verbose logging. Useful for figuring out why authentication isn't
# working.
#auth_verbose = no

# Even more verbose logging for debugging purposes. Shows for example SQL
# queries.
#auth_debug = no

# digest-md5 authentication process. It requires special MD5 passwords which
# /etc/shadow and PAM doesn't support, so we never need roots to handle it.
```

Note that the passwd-file is opened before chrooting and dropping root
privileges, so it may be 0600-root owned file.

#auth = digest_md5
#auth_mechanisms = digest-md5
#auth_realms =
#auth_userdb = passwd-file /etc/passwd.imap
#auth_passdb = passwd-file /etc/passwd.imap
#auth_user = imapauth
#auth_chroot =

if you plan to use only passwd-file, you don't need the two auth processes,
simply set "auth_methods = plain digest-md5"

680 ☆mail 日志

Sendmail 很好的利用的它的日志文件，你可以追捕到大部分问题的所在。
mail.log 和 mail.err 是你需要注意的两个主要文件，还有就是 ail.warn 和 mail.info。
这些文件有许多共同之处，所以我一般只看前两个文件。一般而言，你可以在 /var/log/下找到这些文件，对特定的系统和Linux版本，可能在其他目录下。如果必要的话，你可以察看/etc/syslog.conf文件。典型的电子邮件成功发送一般是这样的:

Oct 25 18:22:14 example sendmail[29322]: SAA29322: from=user, size=193, class=0, pri=60193, nrcpts=2, msgid=<200110260122.SAA29322@panda.com>, relay=user@local

Oct 25 18:22:14 example sendmail[29324]: SAA29322: to=help@techsupport.net, ctladdr=user (500/1000), delay=00:00:00, xdelay=00:00:00, mailer=local, stat=Sent

你可以在 mail.log 看到和下面类似的拒绝信息:

Oct 23 14:23:51 example sendmail[27467]: OAA27467: ruleset=check_rcpt, arg1=, relay=west1.mail-abuse.org [204.152.186.193], reject=550 ... Relaying denied

通常，你所需要的所有信息都存储在这些日志文件中，包括用户名，主机名，出错代码。这些出错代码很有用处，可以通过这些让你对自己有更好的了解。

tail /var/log/maillog

681 ☆Mail 测试

[root@mail root]# mail virtualuser@cngnu.org
Subject: test by me

this is a test.
.
CC:

[root@mail root]# mailq
Mail queue is empty

[root@mail root]# tail /var/log/maillog

使用 mailq 来查看邮件队列是否有错误，并查看/var/log/mail/*是否有错误信息。如果一切正常，说明信件已经发送到 tester 了。

682 ☆测试 pop/imap

测试收信，先测试 POP3:
telnet panda 110

然后输入：
user username
+OK

pass password　　(注意是 pass)
如果 LOGIN 成功，会有提示。
+OK Logged in.

list
+OK 2 messages:
1 849
2 823
.
这样的返回信息说明已经一切正常，可以提供服务了。

STAT
+OK 2 1672

TOP 1
+OK
Return-Path: <student@panda.panda.com>
Received: from panda ([192.168.152.1])
　　by panda.panda.com (8.13.1/8.13.1) with SMTP id k7KJE2DZ004002
　　for <student@panda.panda.com>; Mon, 21 Aug 2006 03:14:02 +0800

Date: Mon, 21 Aug 2006 03:14:02 +0800
From: student@panda.panda.com
Message-Id: <200608201914.k7KJE2DZ004002@panda.panda.com>
To: <student@panda.panda.com>
Subject: =?utf-8?B?TWljcm9zb2Z0IE9mZmljZSBPdXRsb29rIOa1i+ivlea2iOaBrw==?=
MIME-Version: 1.0
Content-Type: text/plain;
	charset="utf-8"
Content-Transfer-Encoding: 8bit
X-IMAPbase: 1154223507 6
Status: O
X-UID: 5
Content-Length: 120
X-Keywords:

DELE 1
+OK Marked to be deleted.

QUIT
+OK Sayonara
Connection closed by foreign host.

再测试 IMAP:

imtest -m login -a panda@panda.panda.com localhost
C: C01 CAPABILITY
S: * OK mail.cngnu.org Cyrus IMAP4 v2.1.16 server ready
S: * CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ NAMESPACE UIDPLUS ID
NO_ATOMIC_RENAME UNSELECT MULTIAPPEND SORT THREAD=ORDEREDSUBJECT
THREAD=REFERENCES IDLE
S: C01 OK Completed
Password:
C: L01 LOGIN virtualuser {6}
+ go ahead
C: <omitted>
L01 OK User logged in
Authenticated.
Security strength factor: 0

  . select inbox
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)

* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen *)]
* 1 EXISTS
* 1 RECENT
* OK [UIDVALIDITY 1021736432]
* OK [UIDNEXT 3]
  . OK [READ-WRITE] Completed

  . fetch 1:1 (FLAGS BODY[HEADER.FIELDS (DATE FROM)])
* 1 FETCH (FLAGS (\Seen) BODY[HEADER.FIELDS (DATE FROM)] {68}
Date: Mon, 20 May 2002 09:26:09 +0800 (CST)
From: wxy@cngnu.org

)
  . OK Completed
  . logout
* BYE LOGOUT received
  . OK Completed

683 ☆procmail

邮件分拣
Linux 的 sendmail 使用 procmail 作为信件的最终投递代理。这个程序有一些非常有用的 ，对于我们来说，最重要的功能是信件的自动过滤和分拣功能。

信件分拣大概是用户最希望的功能，它按照邮件的文件头(发信地址，收信地址等等)甚至邮件的正文进行归类，并且可以自动存放在各个文件中或者转发给别的用户账号 。马上我们会看到，用 procmail 配置自动分拣是一件非常容易的事情。

通常的 sendmail 配置中已经使用了 procmail 作为邮件最终投递代理，如果你的 sendmail 已经改乱了，可以使用 FEATURE(local_procmail)设置这个功能。

procmail 主要依靠用户宿主目录下面的.procmailrc 中的信息来处理邮件.如果这个配 制文件不存在,则 procmail 只是简单的将邮件保存到用户的信箱中.

一般来说，.procmailrc 文件由配置行和行为规则组成，规则的一般格式是
0 选项
[零个或多个条件,每个一行]
[动作命令] [动作命令]
0 表示开始一条规则，后面可以加上一些单字符的选项，选项主要有下面的一些:
H 搜索匹配邮件头部
B 搜索匹配整个邮件
D 匹配时区分大小写

A 如果前面最近的一个没有 A 或 a 选项的规则执行,执行本规则
a 如果上面一条规则执行,则本规则执行
E 同 A 相反,前面最近的没有 E 或 e 选项的规则没有执行,则执行本规则
e 同 a 相反
h 通过管道传送邮件首部(缺省)
b 通过管道传送邮件主体(缺省)
c 复制一个邮件
I 忽略所有写操作中的错误
r 原始模式,即 procmail 不对 mail 进行任何模式的处理
条件用一个*号开始，后面跟上正则表达式
动作命令就是 procmail 在规则成功之后使用的命令，一般的命令有下面一些:

{}
开始一个语句段，表示把一组命令集合成一个动作。如果你要嵌套处理规则，(例如 ，对于来自 test 的邮件，区分是含有 computer 字符串还是含有 physics 字符串)那么必须 使用语句段把子规则括起来。

!
转发信件给某个用户

|
启用管道将邮件传送给后面的程序，例如|auto-reply 表示启动 auto-reply 程序并且将 邮件内容作为标准输入传递给它。

任意文件名
将邮件存入某个文件。如果文件已经存在，就添加在文件的末尾。

对于熟悉 perl 或 C 语言的的用户，很容易用 procmail 的管道功能做出邮件的自动回复程序，这里不再介绍了，想进一步了解 procmail 的用户可以用

684 ☆使用 POP 客户端

所有的现在的邮件用户代理（MUA），例如 netscape,elm,Outlook,pine 和 mutt 都是使用 POP 的，可以被用作 POP 的客户端。每一个的配置都有所不同。同样有一个流行字符界面的的 POP 客户端叫做 fetchmail。fetchmail 是高度的可配置的，可以查询多个邮箱，可以作为守护进程运行，这样使得其每五分钟查询用户的邮箱。fetchmail 在主机上递送邮件到邮件传送代理（MTA），例如 sendmail。我们将勾画出以后如何安装 fetchmail 和使用其来查询我们装过的 POP 服务器。

注意到有很多选项可以影响 fetchmail 的行为。建立一个~/.fetchmailrc 文件如下所示:

~student/.fetchmailrc

poll stationX.exmaple.com with protocol pop3: user studentXX there is user studentXX here password "password"

由于密码存储在该文件中，因此 fetchmail 将会拒绝运行除非您把该文件的属性设定为对于仅仅文件的所有者只读。注意还可以使用 chown 改变由 root 创建的文件的所有者为 studentXX。

chmod 600 ~student/.fetchmailrc
chown student.student ~student/.fetchmailrc

尝试使用 studentXX 登陆到 POP3 邮箱

echo "hello student" | mail -s "Hola" student
su - student
fetchmail -v
exit

fetchmail 能不能接收到 student 的 POP 邮件？将递送 student 的邮件到哪里？比从本地获取 POP 邮件有意义么？

让您的伙伴在另外一台机器上建立相同的~/.fetchmailrc 文件（或者配置其它诸如 mozilla 的 MTA）试图从您的服务器上进行收信。

685 ☆fetchmail

许多用户有多个邮件账户，有些在你的管理范围之内，有些在别的服务器上面。管理 所有这些邮件是非常恶心的事情。另外，也许你的系统仅仅是一个拨号代理(参考第八 章)，不可能始终接在 internet 上面。你需要的是在系统连接到 internet 的时候发出电 子邮件，同时自动去接收电子邮件。在国内，典型的做法是每人申请一个本地电子邮件 账户和一个免费电子邮件账号，问题就是，如何从另外的 ISP 提供的电子邮件账号哪里自 动地接收邮件？

当然你可以让你的用户自己解决这个问题，不过这种僵硬的方法不见得合适。一般我 们采用另外的办法，就是 fetchmail，它是一个自动的邮件接力程序，可以让它从远程的 pop3 账号处取得邮件，然后扔进用户的本地邮箱。(关于 pop3 的情况参考下一节)

可以直接用命令行调用 fetchmail:

$ fetchmail -p [协议] -u [用户名] [服务器]

fetchmail 得到的信息将直接投入 localhost 机器，账号是你启动 fetchmail 时使用的账 号。

$ fetchmail -p POP3 -u yuanban mail.asnc.edu.cn

Enter password for yuanban@mail.asnc.edu.cn:

使用 pop3 协议从 mail.asnc.edu.cn 取得 yuanban 的信件。fetchmail 要求你输入 yuanba n 的密码，然后进行验证，成功的话会出现下面的信息:

1 message for yuanban at mail.asnc.edu.cn (551840 octets).

reading message 1 of 1 (551840 octets) ....d

表示已经完成了一封信件的转交。

对于更复杂的情况，建议你使用 fetchmail 的配置文件，即用户宿主目录下面的.fetc hmailrc。这个文件的详细资料可以参考 fetchmail 的文档，这里只用一个简单地例子来 介绍它。我们看一个简单的.fetchmailrc 范例:

set syslog

set postmaster "isee"

poll 202.96.44.11 with proto POP3 and options

envelope Delivered-To:

user "mere" there with password "xxxxxxx" is isee@snail.home here

no keep

user "isee" there with password "yyyyyyy" is isee@snail.home here

no keep

这个范例相当简单。fetchmail 在读取配置文件的时候会忽略所有的"there","here", "and","with","has","wants","options"之类的单词。

第一行 set syslog 程序定义 fetchmail 的记录文件使用系统记录功能。set postmaste r 设置的是出错时的管理员账号，接下来的 poll 行定义了 fetchmail 使用的协议为 POP3， 连接的服务器是 202.96.44.11。

envelope Delivered-To:这个选项比较有趣，它在信封上加上一个 Delivered-To 说明 ，这主要是为了让 fetchmail 可以把信件投递到一些使用 qmail(见下面)的系统中。

接下来是 fetchmail 的主体部分，一个 user 的行定义了一个接力方法，现在的定义是取 得 202.96.44.11 服务器上 mere 账号的信件，取信的口令是 xxxxxxx，然后投递到 isee@sn ail.home，同样，isee 的信件也被自动投递到 isee@snail.home。no-keep 选项表示不在 原始服务器上保留邮件。

下面是配置 fetchmail 的一些常用选项:

set logfile 制定 log 文件

set syslog 使用系统 log 文件

via 指定 DNS 机器名来取代 poll 中的机器明

proto 指定协议

port 指定端口

timeout 指定超时时间

interface 指定网络界面

user 指定远程用户

is 将本地用户和远程用户联系起来

to 同 is

pass 口令

preconnect 连接开始前执行的外部命令

postconnect 连接结束后执行的外部命令

keep 在服务器上保留邮件备份

no keep 不保留邮件备份

在配置了.fetchmailrc 之后，只要直接执行 fetchmail 就可以实现自动的信件投递了。 实际上，你甚至可能使用 fetchmail 作为一个 daemon 程序来实现自动的邮件转交，或是设 置 fetchmail 让它支持 qmail 的虚拟域。

在 fetchmail 的发行版本中还提供了一个图形化的 fetchmail 配置程序,称为 fetchmail conf。如果你对 fetchmail 的语法感到困难，只要使用这个程序就可以进行配置。启动这 个程序的方法很简单，直接在 xterm 下面执行 fetchmailconf &:

图 7.1 配置 fetchmail

选择 configure fetchmail 出现:

图 7.2 配置 fetchmail(2)

只要选择 Novice Configuration，然后出现服务器设置选单:

图 7.3 配置 fetchmail(3)

在 New Server 中加入你准备收信的服务器，然后回车，服务器名字就会出现在列表框 中，双击服务器名字，编辑关于服务器的设置:

图 7.4 配置 fetchmail(4)

设置服务器的类型和服务器上的账户名字，双击账户名字并且输入对应的密码以及和 本地账户之间的对应关系，OK 退出就可以了

686 ☆squirrelmail

rpm -q squirrelmail

squirrelmail 的主配置文件为/etc/squirrelmail/config.php

更改这个配置用/usr/share/squirrelmail/config/conf.pl
更方便

687 ☆SMB 协议

SMB(Server Message Block，服务信息块)协议可以看作是局域网上的共享文件/打印机的一种协议，它可以为网络内部的其它 Windows 和 Linux 机器提供文件系统、打印服务或是其他一些信息。SMB 的工作原理是让 NetBIOS(Win95 网络邻居通信协议)与 SMB 这两种协议运行在 TCP/IP 的通信协议上，且使用 NetBIOS nameserver 让用户的 Linux 机器可以在 Windows 的网络邻居里被看到，所以就可以和 Win95/NT 主机在网络上相互沟通，共享文件与服务了。
目前 Microsoft 正在开发一种新的文件和打印共享协议--CIFS(Common Internet Files System，通用网络文件协议)，该协议支持 TCP/IP 和 DNS 等协议，能在 www 上支持文件和打印共享。在 CIFS 下服务器实际上是 DNS 名，由主机名和域名组成。这种变化是从 NetBIOS 命令结构中分离出来

的。目前 CIFS 还没有得到广泛的应用。现在在 Linux 和 Win95/NT 之间的文件共享注意还是 SMB 和 NFS。

688☆Samba

Samba 是用来实现 SMB 的一种软件，由澳大利亚的 Andew Tridgell 开发，是一种在 Linux(Unix)环境下运行的免费软件。

在 NetBIOS 出现之后，Microsoft 就使用 NetBIOS 实现了一个网络文件/打印服务系统，这个系统基于 NetBIOS 设定了一套文件共享协议，Microsoft 称之为 SMB(Server Message Block)协议。这个协议被 Microsoft 用于它们 Lan Manager 和 Windows NT 服务器系统中，实现不同计算机之间共享打印机、串行口和通讯抽象(如命名管道、邮件插槽等)。
随着 Internet 的流行，Microsoft 希望将这个协议扩展到 Internet 上去，成为 Internet 上计算机之间相互共享数据的一种标准。因此它将原有的几乎没有多少技术文档的 SMB 协议进行整理，重新命名为 CIFS(Common Internet File System)，并打算将它与 NetBIOS 相脱离，试图使它成为 Internet 上的一个标准协议。
因此，为了让 Windows 和 Unix 计算机相集成，最好的办法即是在 Unix 计算机中安装支持 SMB/CIFS 协议的软件，这样 Windows 客户就不需要更改设置，就能如同使用 Windows NT 服务器一样，使用 Unix 计算机上的资源了。samba 是用来实现 SMB 的一种软件，它的工作原理是，让 NETBIOS(Windows95 网络邻居的通讯协议)和 SMB(Server Message Block)这两个协议运行于 TCP/IP 通信协议之上，并且使用 Windows 的 NETBEUI 协议让 Unix 计算机可以在网络邻居上被 Windows 计算机看到。它的功能有:
文件服务和打印服务(在 Linux 和 Win95/NT 之间系统之间提供打印机和磁盘的共享)
1.共享 Linux 磁盘给 Win95/NT
2.共享 Win95/NT 磁盘给 Linux 机器
3.共享 Linux 打印机给 win95/NT
4.共享 win95/NT 打印机给 Linux 机器。
登陆服务器，使用 Windows 客户能注册到网络上
作为主要域控制器和域中成员的功能
WINS 服务器以及浏览功能
支持 SSL(Secure Socket Layer)
支持 SWAT (Samba Web Administration Tool)
Samba 除了支持 Linux(Unix)和 Win95/NT 之外，还支持 DOS、IBM OS/2、Macintosh 等操作系统。

同时它的文件服务功能比 NT 系统还高，而且在 Windows2000 之前就提供了用户磁盘空间限制的功能。
Samba Server 建立了 Linux 与 Windows 环境的沟通管道，也可以做为 Print Server 提供 Windows 远程联机打印;若是使用 Samba Server 搭配 Apache Web Server，可在 Windows 环境下藉由『网上邻居』登入 Linux 主机里使用者的个人帐号放置网页目录(当然是在局域网络内才行)，有了这项功能，编辑个人网页就如同在本机操作一般方便。除此之外，Samba Server 也可以完全取代 NT/2000 PDC(Primary Domain Controler)成为 NT/2000 网域主控者管理 NT/2000 网域机群，当然，Samba 也可以将目录、文件分享给其它 Unix Like、Mac、OS/2 的机器使用，应用层面可以

说是相当广阔，更令人惊讶的是 Samba Server 也可做为 WINS Server，若配合 DHCP Server 更可以管理大型 NT/2000 网域.

Samba 应该范围主要是 Windows 和 Linux 系统共存的网络中使用;如果一个网络环境都是 Linux 或 Unix 类的系统，没有必要用 Samba，应该用 NFS 更好一点;

Samba 有两个服务器，一个是 smb，另一个是 nmb
守护进程 smbd 和 nmbd 是 Samba 的核心，在全部时间内运行。

Smbd 守护进程在 SMB 软件包到达网上时对它们进行处理，并且为使用或共享它的资源与 Linux 进行协调。主要用来处理文件分享和打印分享服务
smb 是 Samba 的主要启动服务器，让其它机器能知道此机器共享了什么;

nmbd 程序使得通过企图计算机可以浏览 Linux 服务器。用来处理 WINS 名称解析服务及 NT Browser Service(即网上邻居)
nmb 是解析用的，解析了什么呢？就是把这台 Linux 机器所共享的工作组及在此工作组下的 netbios name 解析出来;

如果不打开 nmb 服务器的话，只能通过 IP 来访问，比如在 Windows 的 IE 浏览器上打入下面的一条来访问;
\\192.168.1.5\共享目录
\\192.168.1.5\opt

689 ☆安装 samba

samba-common-2.2.7a-7.9.0
samba-client-2.2.7a-7.9.0
samba-swat-2.2.7a-7.9.0
samba-2.2.7a-7.9.0

690 ☆Samba 使用前配置

Samba 用的 netbios 协议，如果您用 Samba 不成功，
先确定在 /etc/services 文件里面这些句子没有被批注掉:
netbios-dgm 138/tcp # NETBIOS Datagram Service
netbios-dgm 138/udp
netbios-ssn 139/tcp # NETBIOS session service
netbios-ssn 139/udp

然后是建立 /etc/lmhosts 文件(如果它不存在的话),建立起各主机的对应。我们只需将 MS Windows 主机对应建立起来则可,但一定要包括作为 SAMBA 服务器的 Linux 主机本身:(类似于/etc/hsots)

vi /etc/lmhosts
192.168.152.128 panda
192.168.152.129 askpanda

691 ☆Samba 防火墙

有时你的防火墙可能会把 smbd 服务器的端口封掉，所以我们应该 smbd 服务器所占用的端口;下面查看中，我们知道 smbd 所占用的端口是 139 和 445 ;
[root@localhost ~]# netstat -tlnp |grep smb
tcp 0 0 0.0.0.0:139 0.0.0.0:* LISTEN 10639/smbd
tcp 0 0 0.0.0.0:445 0.0.0.0:* LISTEN 10639/smbd

692 ☆samba 启动

service smb start

开机自动激活 samba
#chkconfig --level 345 smb on

693 ☆Samba 配置

system-config-samba

694 ☆samba-swat

/etc/xinetd.d/swat
service swat
{
port = 901 (swat 用到的端口是 901)
socket_type = stream(类似 tcp 协议的东西,呵呵这是我自己的理解)
wait = no
only_from = 127.0.0.1(只从这个启动 swat,关键!)
user = root(启动 swat 用到的名子)
server = /usr/sbin/swat
log_on_failure += USERID
disable = yes (一定要把 yes 改成 no)
}

重启启动 xinetd 就可以了

然后在浏览器里些上:http://127.0.0.1:901/
(注意,这个 IP 地址要跟你在 SWAT 配置文件里的 IP 地址一样才形,切记!)

695 ☆测试 Samba

smbclient -L localhost -N

696 ☆smb.conf

smbd 和 nmbd 这两个守护进程启动时(通常为系统引导时)读配置文件/etc/samba/smb.conf，这一配置文件向这两个守护进程说明输出什么共享、共享输出给谁以及如何进行输出等等。

在安装完 Samba 后，还需要定制它的配置文件 smb.conf，才能使 Samba 正常工作以符合要求。smb.conf 文件的的语法结构与 Windows 的*.ini 文件十分类似;文件结构主要包括三部分:全局参数部分、目录共享部分、打印共享部分

在 Samba 的软件包里的 example 目录中有一个缺省的配置文件 smb.conf.defaults。我们可以对它做适当修改后拷贝到你的安装目录下的 lib 目录里，并改名为 smb.conf。清单 1-2 是 Samba 系统提供的标准的配置文件。为方便读者阅读，本文翻译了文件中的注释并适当的添加了解释。

文件被分隔成若干节，每一节都由一个被方括号括起来的标识开始(例如，[global]、[home]、[printers])，每一个配置参数或是一个全局参数(影响或控制整个服务器)，或是一个服务参数(影响或控制服务器提供的某项服务　)。
global 部分定义的参数用来控制 Samba 的总特性。除 global 部分外，每一部分都定义了一个专门的服务。

你可以使用下面的语句来指定一个参数:
name=VALUE
name 可以是一个单词或者用空格隔开的多个单词。VALUE 可以是布尔值(ture 或 false;yes 或 no;1 或 0)、数字或字符串。

注释以分号开头，可以单独一行，也可以跟在一条语句之后。

通过在一行的最后一个字符后加反斜杠"\"可以将一行分成多行。

每一部分的名字和参数都不区分大小写，例如，参数 browseable=yes 与 browseable=YES 是完全等价的

标准 smb.conf 文件
这是服务器的主要配置文件。您应该阅读 smb.conf(5)的用户手册以了解下面列出的每一个选项。Samba 有很多的选项，它们之中的大多数并没有出现在这个例子中。
#
以分号";"或井号"#"开始的每一行都是注释，在执行时被忽略。在本例中我们使用"#"作为注释而使用";"作为可选配置的注释。

#
注意:无论何时修改了这个配置文件，您都要运行"testparm"命令来检查您所做的修改有没有基本的语法错误。
#
这里说一下 samba 定义的变量:
%S = 当前服务名(如果有的话)
%P = 当前服务的根目录(如果有的话)
%u = 当前服务的用户名(如果有的话)
%g = 当前用户说在的主工作组
%U = 当前对话的用户名
%G = 当前对话的用户的主工作组
%H = 当前服务的用户的 Home 目录
%v = samba 服务的版本号。
%h = 运行 samba 服务机器的主机名
%m = 客户机的 NETBIOS 名称
%L = 服务器的 NETBIOS 名称
%M = 客户机的主机名
%N = NIS 服务器名
%p = NIS 服务的 Home 目录
%R = 说采用的协议等级(值可以是 CORE, COREPLUS, LANMAN1, LANMAN2，NT1)
%d = 当前服务进程的 ID
%a = 客户机的结构(只能识别几项:samba，WfWg，WinNT，Win95)
%I = 客户机的 IP
%T = 当前日期和时间

#=================== 全局变量设置 Global Settings ====================
[global]
# workgroup = NT-Domain-Name or Workgroup-Name。缺省的组名是 MYGROUP。不分大小写
# workgroup 用来指定您的机器在网络上所属的 NT 域名或组名。格式是
workgroup = MYGROUP

; netbios name = CCGD.COM
#netbios 名字;就是在 Windows 中显示出来的计算机名

; netbios aliases =
#设置 samba 服务的别名，(netbios 的别名，在网络里同 netbios 一样做用);

; netbios scope =
#

server string 用来设置 NT 描述域。缺省值是 Samba Server 。
server string = Samba Server

#设定机器的描述，当我们通过网络邻居访问的时候可以在备注里面看见这个内容，而且还可以使用 samba 设定的变量。

下面的选项对于安全很重要。它允许您设置哪些领域的机器可以访问您的 Samba 服务器。下面的这个例子允许两个 C 类子网和"lookup"的连接请求而禁止来自其他网段机器的连接请求。有关的例子请参看 smb.conf 的用户手册。(网络注意后面加"."号，各个项目间用空格隔开，记得把本机也加进去)
; hosts allow = 192.168.1. 192.168.2. 127.

设定是否自动共享打印机而不用设置下面的[printer]一节的相关东西
允许自动加载打印机列表，而不需要您单独设置每一台打印机。
load printers = yes

到 printcapFile(一般是/etc/printcap)这个文件中取得打印机的描述信息
您也许希望覆盖原有的 printcap 文件。
; printcap name = /etc/printcap
对于 SystemV 系统，如果将 printcap 名设置为 lpstat 将允许您从 SystemV 的
spool 中自动获得打印列表。
; printcap name = lpstat

除非您的打印机不是标准型号，否则您没有必要在下面指定打印机系统的类型。
定义打印系统的类型，缺省是 lprng,目前支持的打印机系统包括:bsd, sysv, plp, lprng, aix, hpux, qnx
; printing = bsd

定义游客帐号，而且需要把这个帐号加入/etc/passwd，不然它就用缺省的 nobody
如果希望建立一个客户帐号，去掉下面语句前面的分号";"。同时，您必须在/etc/passwd 中加入这个帐号的定义，否则将使用用户"nobody"作为客户帐户。
; guest account = pcguest

定义日志文件的位置 LogFileName(一般是用/var/log/samba/%m.log)
此选项将为每一个与服务器连接的机器定义一个单独的日志文件。
log file = /var/log/samba/%m.log

定义日志文件的大小 size(单位是 KB，如果是 0 的话就不限大小)
max log size = 50

; min passwd length = 5
#设置密码的最小长度;

; null passwords = No
#是否使用空密码;

# 定义安全模式。大多数人都喜欢用户级安全模式，详细内容参看 security_level.txt
security = user
# 定义 samba 的安全级别，按从低到高分为四级:share，user，server，domain。它们对应的验证方式如下:
# share:没有安全性的级别，任何用户都可以不要用户名和口令访问服务器上的资源。
# user:samba 的默认配置，要求用户在访问共享资源之前资源必须先提供用户名和密码进行验证。
# server:和 user 安全级别类似，但用户名和密码是递交到另外一个服务器去验证，比如递交给一台 NT 服务器。如果递交失败，就退到 user 安全级。
# domain:这个安全级别要求网络上存在一台 Windows 的主域控制器，samba 把用户名和密码递交给它去验证。
# 后面三种安全级都要求用户在本 Linux 机器上也要系统帐户。否则是不能访问的。

只有当安全模式设置为服务器级(security = server 或 domain)时，才定义下面选项。
; password server = <NT-Server-Name>

; password level = n
这是设定针对一些 SMB 客户像 OS/2 之类而设的，这样的系统在发送用户密码的时候，会把密码转换成大写再发送，这样就和 samba 的密码不一致，这个参数可以设定密码里允许的大写字母个数，这样 samba 就根据这个数目对接收到的密码进行大小写重组，以重组过的密码尝试验证密码的正确性。n 越大，组合的次数就越多，验证时间就越长，安全性也会因此变得越低。例如 n=2，用户的密码是 abcd，但发送出去其实是 ABCD，samba 就会把这个 ABCD 进行大小写重组，组合后的结果可以是: Abcd, aBcd, abCd, abcD, abcd, ABcd, AbCd, AbcD,aBCd,aBcD,abCD。
所以如果没有必要，就把 n 定为是零。这样的话 samba 只尝试两次，一个是接收到的密码，另一个尝试的是这个密码都是小写的情况。

; username level = n
这个是对于用户名的情况，说明和上面一项类似。

如果用户想使用加密口令的话，请参阅 ENCRYPTION.txt、Win95.txt 和 WinNT.txt 文件，请在阅读以上文件后使用下面选项。
; encrypt passwords = yes
设置是否对密码进行加密，samba 本身有一个密码文件/etc/samba/smbpasswd，如果不对密码进行加密则在验证会话期间客户机和服务器之间传递的是明文密码，samba 直接把这个密码和 Linux 里的/etc/samba/smbpasswd 密码文件进行验证。但是在 Windows 95 OS/R2 以后的版本和 Windows NT SP3 以后的版本缺省都不传送明文密码，要让这些系统能传送明文密码必须在其注册表里更改，比较麻烦，好的方法就是把这里的这个开关设置为 yes。

; root directory =
#设置 root 访问时的主目录，系统默认是不允许 root 进行访问的;

; smb passwd file = smbPasswordFile

设置存放 samba 用户密码的文件 smbPasswordFile(一般是/etc/samba/smbpasswd)。

; ssl CA certFile = sslFile
当 samba 编译的时候支持 SSL 的时候，需要指定 SSL 的证书的位置(一般在 /usr/share/ssl/certs/ca-bundle.crt)。

; unix password sync = yes|no
设定 Linux 与 samba 使用相同的密码
; passwd program = /usr/bin/passwd %u
; passwd chat = *New*UNIX*password* %n*ReType*new*UNIX*password* %n*passwd:*all*authentication*tokens*updated*successfully*
这三项设置能否从 windows 的应用程序修改 unix 系统的用户密码

; username map = UsermapFile
指定用户映射文件(一般是/etc/samba/smbusers)，当我们在这个文件里面指定一行 root = administrator admin 的时候，客户机的用户是 admin 或者 administrator 连接时会被当作用户 root 看待。

; time server = No
#设置成时间服务器

指定对不同机器的连接采用不同的配置文件 MachineConfFile(一般为了灵活管理使用 /etc/samba/smb.conf.%m，由于采用了 samba 的变量，把配置文件和客户机的 NETBIOS 名称关联起来，能很容易地控制这些客户机的权限和设置)。
使用此选项允许您对每一个机器使用不同的配置。%m 将被替换成与服务器请求连接机器的 NetBIOS 名。
; include = /usr/local/samba/lib/smb.conf.%m

大多数人会发现此选项将显著提高服务器的执行效率，请参读 speed.txt 和用户手册以了解更多细节。
这个是网络 socket 方面的一些参数，能实现最好的文件传输性能。相关的选项还有 SO_KEEPALIVE、 SO_REUSEADDR、SO_BROADCAST、IPTOS_LOWDELAY、IPTOS_THROUGHPUT、SO_SNDLOWAT(*)、SO_RCVLOWAT(*)，带*号的要指定数值。一般如果在本地网络，就只用 IPTOS_LOWDELAY,如果是有一个本地网络的，就用 IPTOS_LOWDELAY TCP_NODELAY，如果是广域网络，就试试 IPTOS_THROUGHPUT。
; socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
socket options = TCP_NODELAY

# 如果有多个网络接口，就必须在这里指定.
interfaces=eth0
interfaces=192.168.16.177

interfaces=192.168.16.177/24
interfaces=192.168.16.177/255.255.255.0
; interfaces = 192.168.12.2/24 192.168.13.2/24

; remote browse sync = host(subnet)
这里指定浏览列表同步信息从哪里取得， 如果用 host(比如 192.168.3.25)或者整个子网(192.168.5.255)。
这里说明一下什么是浏览(Browse):
在 SMB 协议中，计算机为了访问网络资源，就需要了解网络上存在的资源列表(例如在 Windows 下使用网络邻居查看可以访问的计算机)，这个机制就被称为浏览(Browse)。虽然 SMB 协议中经常使用广播的方式，但如果每次都使用广播的方式了解当前的网络资源(包括提供服务的计算机和各个计算机上的服务资源)，就需要消耗大量的网络资源和浪费较长的查找时间，因此最好在网络中维护一个网络资源的列表，以方便查找网络资源。只有必要的时候，才重新查找资源，例如使用 Windows 下的查找计算机功能。
但没有必要每个计算机都维护整个资源列表，维护网络中当前资源列表的任务由网络上的几个特殊计算机完成的，这些计算机被称为 Browser，这些 Browser 通过记录广播数据或查询名字服务器来记录网络上的各种资源。
Browser 并不是事先指定的计算机，而是在普通计算机之间通过自动进行的推举产生的。不同的计算机可以按照其提供服务的能力，设置在推举时具备的不同权重。为了保证一个 Browser 停机时网络浏览仍然正常，网络中常常存在多个 Browser，一个为主 Browser(Master Browser)，其他的为备份 Browser。

; remote announce = host(subnet)
指定这些机器向网络宣告自己，而不是有 Browser 得到。

这个参数指定 nmbd 是否试图成为本地主浏览器，默认值是 yes，如果设为 no 则 samba 服务器就永远都不会成为本地主浏览器。但即使设置了 yes，也不等于 samba 服务器就会成为本地主浏览器。只是参与本地主浏览器选择。
#如果不想使您的 Samba 服务器成为局域网内部的主浏览服务器，将此选项设为 no
; local master = no

n 的值是个整数，决定了 nmbd 是否有机会成为本地广播区域的工作组里的本地主浏览器，默认值是零，零则意味着 nmbd 失去浏览选择。如果要 nmbd 更有机会成为本地主浏览器的话，可以设为 65。
OS Level 决定了该服务器在局域网内的访问优先权。
; os level = 33

这个参数让 nmbd 成为一个域浏览器，取得各本地主浏览器的浏览列表，并将整个域的浏览列表递交给各本地主浏览器
Domain Master 将 Samba 服务器定义为主域浏览器。此选项将允许 Samba 在子网列表中比较浏览。如您已经有一台 Windows NT 域控制器，不要使用此选项
设定 samba 成为 PDC (网域主控者),注意:若是将 Samba 设定为独立服务器，则无须设定此项。

; domain master = yes

Preferred Master 使 Samba 在启动时选择一个本地浏览器并给它获得选择的较高的机会
这个参数指定 nmbd 是否是工作组里的首要的主浏览器，如果指定为 yes，nmbd 在启动的时候就强制一个浏览选择。指定该参数为 yes 时最好把 domain master 也指定为 yes。用这个参数的时候要注意的是在 samba 服务器所在的子网上如果有其它的机器(不管是 WINDOWS NT 还是另一个 samba 服务器)也指定为首要的主浏览器时，这些机器都会因为争夺主浏览器而在网络上广播，引起不必要的网络性能下降。
; preferred master = yes

#Domain master 和 local master
工作组和域这两个概念在进行浏览时具备同样的用处，都是用于区分并维护同一组浏览数据的多个计算机。事实上他们的不同在于认证方式上，工作组中每台计算机都基本上是独立的，独立对客户访问进行认证，而域中将存在一个(或几个)域控制器，保存对整个域中都有效的认证信息，包括用户的认证信息以及域内成员计算机的认证信息。浏览数据的时候，并不需要认证信息，Microsoft 将工作组扩展为域，只是为了形成一种分级的目录结构，将原有的浏览和目录服务相结合，以扩大 Mircrosoft 网络服务范围的一种策略。
工作组和域都可以跨越多个子网，因此网络中就存在两种 Browser，一种为 Domain Master Browser ，用于维护整个工作组或域内的浏览数据，另一种为 Local Master Browser，用于维护本子网内的浏览数据，它和 Domain Master Browser 通信以获得所有的可浏览数据。划分这两种 Browser 主要是由于浏览数据依赖于本地网广播来获得资源列表，不同子网之间只能通过浏览器之间的交流能力，才能互相交换资源列表。
但是，为了浏览多个子网的资源，必须使用 NBNS 名字服务器的解析方式，没有 NBNS 的帮助，计算机将不能获得子网外计算机的 NetBIOS 名字。Local Master Browser 也需要查询 NetBIOS 名字服务器以获得 Domain Master Browser 的名字，以相互交换网络资源信息。
由于域控制器在域内的特殊性，因此域控制器倾向于被用做 Browser，主域控制器应该被用作 Domain Master Browser，他们在推举时设置的权重较大。

#仅当您的网络中有一台在安装时设置为主域控制器的 NT 服务器时使用此选项。
; domain controller = <NT-Domain-Controller-SMBName>

启用域登录.如果想使 Samba 成为 Windows95 工作站的登录服务器，则使用此选项。
; domain logons = yes

如果允许域登录服务，那么您也许希望每台机器或每个用户的登录脚本运行一个特定的每工作站的登录批处理文件。
; logon script = %m.bat

运行一个特定的每用户名登录批处理文件。(构建虚拟 NT 域时所需);
; logon script = %U.bat

设置登录用户配置文件,放置 roving profiles 文件的位置(仅用于 Win95 和 WinNT) ，%L 代表该服务器 NetBIOS 名，%U 是用户名，您必须取消后面定义的[Profiles]前面的注释号。
; logon path = \\%L\Profiles\%U

将 samba 设置成 WINS 服务器,Windows 的 Internet 名服务支持记录部分 WINS Support 告诉 NMBD 守护进程支持 WINS 服务器。
; wins support = yes

WINS Serve 选项告诉 NMBD 守护进程作为 WINS 的客户机。
注意:Samba 既可以作为 WINS 服务器也可以作为 WINS 客户机，但不能兼而有之。
; wins server = w.x.y.z

WINS Prox 代表一个非 WINS 客户通知 Samba 响应名字解析请求。要使此选项正常工作必须保证网络中至少有一台 WINS 服务器。缺省值是 NO。
; wins proxy = yes

DNS Proxy 选项决定 Samba 是否通过 DNS 的 nslookups 去解析主机的 NetBIOS 名。对于 1.9.17 以前的版本内置值是 yes ，对于 1.9.18 之后的版本内置值是 no 。
dns proxy = no

; preserve case = yes|no
; short preserve case = yes|no
指定拷贝 DOS 文件的时候保持大小写，缺省是 no

; default case = lower|upper
所有的 DOS 文件的缺省是大写还是小写

; case sensitive = yes|no
大小写敏感，一般是 no,不然会出现一些问题。

; client code page = 950
如果想使用者能看到中文文件名称,可以将这行加进 [global] 设定中
#设置 samba 所使用的字符集(默认是:850，936 是简体中文,默认是 850,936 是简体中文)

status = Yes
#samba 的运行状态;

#================== 定义共享服务 Share Definitions ==================
每个 SMB 服务器能对外提供文件或打印服务，每个共享资源需要被给予一个共享名，这个名字将显示在这个服务器的资源列表中。如果一个资源的名字的最后一个字母为$，则这个共享名就为隐藏共享，不能直接表现在浏览列表中，而只能通过直接访问这个名字来进行访问。在 SMB

协议中，为了获得服务器提供的资源列表，必须使用一个隐藏的资源名字 IPC$来访问服务器，否则客户无法获得系统资源的列表。

[homes]，在 smb.conf 文件中一般没有对这个目录的设定特定内容比如路径等。当客户机发出服务请求时，就在 smb.conf 文件的其它部分查找友特定内容的服务。如果没有发现这些服务，并且提供了 homes 段时，那么就搜索密码文件得到用户的 Home 目录。通过 Homes 段，samba 可以得到用户的 Home 目录并使之共享。

所有使用者的 home 目录
[homes]
comment = Home Directories
# 当一个客户程序以客人用户类出本服务器的共享服务时，不列出 homes 服务。
# 但是本机用户创建的主目录服务仍使用[global]节设定的 browseable。
browseable = no
writable = yes
# browseable 参数控制一项服务是否能够出现在网络资源浏览表中。这里是一些非直觉的东西
# 所谓的使用者 home 目录是指，以使用者帐号登入 samba Server 后个人所拥有的帐号目录，如:以帐号 root 登入后，内定的个人目录是 /root。一般而言，对「使用者 home 目录的设定」着重于是否将此个人目录分享出来及设定存取的权限，其余项目大抵上是无需设定。
# 注意:若您选择了「Public access」、「Writable」两个选项，并不代表所有的使用者皆有权利对您所分享的 home 目录下的文件具有写入权，还得视文件原来在 Linux 下的权限设定，例如，文件权限为 rwxr-xr-x(chmod 755)，表示只有拥有者具可写入权，此点务必留意。

[MyShare]
[]里面的 MyShare 指定共享名，一般就是网络邻居里面可以看见的文件夹的名字。

comment = panda's file
# comment 指的是对改共享的备注

path = /home/panda
# path 指定共享的路径，其中可以配合 samba 变量使用。比如你可以指定 path=/data/%m，这样如果一台机器的 NETBIOS 名字是 grind,它访问 MyShare 这个共享的时候就是进入/data/grind 目录,而对于 NETBIOS 名是 glass 的机器，则进入/data/glass 目录。

allow hosts 和 deny hosts 和前面的全局设置的方法一样这里不再提及

allow hosts = host(subnet)
对登入 Samba 的来源主机做允许登入 Samba 主机
192.168.0.0/255.255.255.0 ;表示允许 192.168.0. class C 登入 Samba。
abc，test，working ;表示允许此三部主机登入 Samba。
@mygroup ;表示允许 mygroup 群组内的成员登入 Samba。

deny hosts = host(subnet)
对登入 Samba 的来源主机做不允许登入 Samba 主机
192.168.0.25 ;表示不允许此部主机登入 Samba。
cracker，badboy ;表示不允许此两部主机登入 Samba。
@mygroup ;表示不允许这个工作群组成员登入 Samba。

writable = yes|no
# writeable 指定了这个目录缺省是否可写，也可以用 readonly = no 来设置可写

user = user(@group)
# user 设置所有可能使用该共享资源的用户，也可以用@group 代表 group 这个组的所有成员，不同的项目之间用空格或者逗号隔开。
# aaa,bbb,ccc,ddd,root ;表示此五个帐号可使用分享目录。
# @cracker ;表示属于 cracker 群组的帐号可使用分享目录。
# @cracker，abc ;表示 cracker 群组成员及 abc 可使用分享目录。

valid users = user(@group)
valid users 指定能够使用该共享资源的用户和组
aaa,bbb,root ;表示此三个帐号登入 Samba。
@cracker ;表示属于 cracker 群组的帐号登入 Samba。
@cracker，abc ;表示 cracker 群组成员及 abc 登入 Samba。

invalid users = user(@group)
invalid users 指定不能够使用该共享资源的用户和组。

read list = user(@group)
read list 指定只能读取该共享资源的用户和组。

write list = user(@group)
write list 指定能读取和写该共享资源的用户和组。

read only = Yes
#设置共享文件或文件夹仅有只读;将允许通过验证的用户对主目录有写入的权限

admin list = user(@group)
admin list 指定能管理该共享资源(包括读写和权限赋予等)的用户和组。

; printer admin =
#设置打印机管理员;

public = yes|no

public 指明该共享资源是否能给游客帐号访问，这个开关有时候也叫 guest ok，所以有的配置文件中出现 guest ok = yes 其实和 public = yes 是一样的。

; guest only = No
#设置共享只允许 GUEST 账号访问;

; guest ok = No
#设置是否允许 GUEST 账号访问;

hide dot files = yes|no
hide dot files 指明是不是像 unix 那样隐藏以"."号开头的文件。

create mode = 0755
create mode 指明新建立的文件的属性，一般是 0755。

create mask = 0744
#设置创建文件时的默认权限;

directory mode = 0755
directory mode 指明新建立的目录的属性，一般是 0755。

directory mask = 0755
#设置创建文件夹时的默认权限;

sync always = yes|no
sync always 指明对该共享资源进行写操作后是否进行同步操作。

short preserve case = yes|no
short preserve case 指明不管文件名大小写。

preserve case = yes|no
preserve case 指明保持大小写。

case sensitive = yes|no
case sensitive 指明是否对大小写敏感，一般选 no,不然可能引起错误。

mangle case = yes|no
mangle case 指明混合大小写

default case = upper|lower
default case 指明缺省的文件名是全部大写还是小写。

force user = panda
force user 强制把建立文件的属主是谁。如果我有一个目录，让 guest 可以写，那么 guest 就可以删除，如果我用 force user= grind 强制建立文件的属主是 grind，同时限制 create mask = 0755，这样 guest 就不能删除了。

force group =
强行设置文件及文件的属主级;

wide links = yes|no
wide links 指明是否允许共享外符号连接，比如共享资源里面有个连接指向非共享资源里面的文件或者目录，如果设置 wide links = no 将使该连接不可用。

max connections = 100
max connections = n 设定同时连接数是 n。

delete readonly = yes|no
delete readonly 指明能否删除共享资源里面已经被定义为只读的文件。

; root preexec =
#设置用户访问共享以 root 身份执行的命令;

; root preexec close = No
#关闭的 ROOT 用户执行 shell 命令;

; root postexec =
#设置当用户退出共享时 root 身份执行的命令;

; available = Yes
#设置共享是否可用;

; volume =
#设置卷标;

有两类特殊的共享，分别是光驱和打印机
光驱的共享设置:
[cdrom]
comment = panda's cdrom
path = /mnt/cdrom
public = yes
browseable = yes
root preexec = /bin/mount -t iso9660 /dev/cd0 /mnt/cdrom

root postexec = /bin/umount /mnt/cdrom
这里 root preexec 指明了连接时用 root 的身份运行 mount 命令，而 root postexec 则指明了断开时用 root 身份运行 umount,有效实现了对光驱的共享。-o iocharset=cp936 中文支持

打印机共享的设置:
[printers]
path = /var/spool/samba
writeable = no
guest ok = yes
printable = yes
printer driver = HP LaserJet 5L
#每一个打印服务必须定义为 printable = yes。
这里 printable 指明该打印机可以打印，
guest ok 说明游客也能打印，
path 指明打印的文件队列暂时放到/var/spool/samba 目录下。
printer driver 的作用是指明该打印机的类型，这样我们在安装网络打印机的时候可以直接自动安装驱动而不必选择。

697 ☆smbusers

注意，账号列表内的用户使用空格隔开。
tom=alarm back

linux 账号=要映射的用户名
root = administrator admin
nobody = guest pcguest smbguest

698 ☆Samba 日志文件

Samba 为 smbd、nmbd 和访问 Samba 的客户提供了下列记录文件，分别记录有关 smbd、nmbd 运行信息和每个客户的访问信息:

Samba 服务的日志文件
Samba 服务的日志默认存放在/var/log/samba 中
为所有连接到 Samba 服务器的计算机建立个别的日志文件(ip 地址.log)，同时也将 NMB 服务和 SMB 服务的运行日志分别写入 nmbd.log 和 smbd.log 日志文件中.

记录 smbd 信息
/var/log/samba/log.smb
记录 nmbd 信息
/var/log/samba/log.nmb
记录客户访问信息

/var/log/samba/log.%m

699 ☆共享访问限制

通过使用不同的参数，可以实现对 Samba 的不同访问控制:通过主机地址限制访问、通过用户口令限制访问、通过用户名限制访问、通过验证读写权限限制访问、通过是否在浏览器列表里显示来控制等。所有这些方法，大大的增强了 Samba 服务器的安全性。

700 ☆通过主机地址限制

使用以下参数来限制通过主机地址访问:(可以在全局和某目录下定义)
allow hosts
hosts allow
deny hosts
hosts deny

如用于[global]段，则应用于所有服务，而忽略在每个服务中的各自的设置。

"allow hosts"、" hosts allow"这两个功能相同的参数指定允许访问一个服务的主机列表，主机列表用"，"号、空格或制表符隔开。

主机列表的组成可以是主机名、IP 地址、子网地址或网络号码/掩码。也可以使"EXCEPT"关键字来限制子网中的个别主机的访问。例如:
hosts allow = 202.204.3 EXCEPT 202.204.3.30
允许在 202.204.3 子网上的主机访问，但禁止其中的 202.204.3.30 访问。

"deny hosts"、" hosts deny"这两个功能相同的参数指定禁止访问某个服务的主机列表，与"allow hosts"相反。

当与"allow hosts"列表冲突时，"allow hosts"列表优先。

701 ☆通过用户口令限制

使用以下参数来限制通过用户口令访问:
security
参数 security 可以确定对客户用户名/口令的验证方式。缺省为"user"，即对 Samba 服务器的任何访问都要通过用户名/口令的验证。
如果客户登录客户机，如 Windows 所用的用户名与 Samba 服务器上的 UNIX 用户名一致，则通过口令验证后就可以访问;如 Windows 所用的口令也与 UNIX 一样，那么不经任何提示就可以访问。

但如果客户机上登录的用户名与 Samba 服务器上的 UNIX 用户名不一致，则需要在一个映射文件中(如/etc/smbusers)写入客户机用户名到 UNIX 用户名的映射，同时使用"username map"参数指定映射文件
username map = /etc/smbusers

在/etc/smbusers 文件中的内容如下:
cuckoo = "cuckoo doo"

这样，在客户机上以"cuckoo doo"注册后，再与 Samba 服务器连接时，输入 UNIX 客户 cuckoo 的口令就可以进入，因为 Samba 已经把"cuckoo doo"映射为 cuckoo。

security 参数的另一个值是"share"，表示 Samba 服务器不需要客户机使用用户名/口令来登录，而是根据每个共享的设置来决定是否需要用户名/口令验证。设置 security 为 share 一般用于客户机上大多数用户名与 Samba 服务器上 UNIX 用户名不相同的情况，以及 Samba 服务器主要提供 guest 访问。

在 smb.conf 文件中，如果没有设置"encrypt passwords"，Samba 将使用 UNIX 的 password 数据库来验证用户口令。这需要客户机将口令以"普通文件"的方式传递过来。
如果设置了"encrypt passwords"，Samba 则使用另一个加密口令文件，通过"smb passwd file"来设置
encrypt passwords = yes
smb passwd file = /etc/smbpasswd

702 ☆通过用户名限制

public
guest ok
上面的两个参数的功能相同，都可以使用相应的服务允许 guest 用户，即不需要用户名/口令验证。如果还指定了"guest only"，那么相应的服务器只允许由 guest 用户来访问。

invalid users
valid users
这两个参数分别设置不允许注册某个服务和允许注册某个服务的用户列表。
一个用户同时出现在两个列表中的时候，不允许注册优先。
列表中的用户名可以用空格隔开，以"@"开头的名字被认为是 NIS 组名或 UNIX 组名;以"+"开头的名字被认为是 UNIX 组名;以"&"开头的名字只被认为是 NIS 组名。

only user
此参数控制是否只允许在"user ="列表中指定的用户访问某个服务。如果设置为 true，则只有在"user ="列表中的用户才能访问。

703 ☆通过读写限制

read only
此参数设置为"yes"，则使用服务的用户不能在该服务的目录中创建或修改文件。

writable
writeable
write ok
这三个参数功能相同。如果它们被设置为"no"，则使用服务的用户不能在该服务的目录中创建或修改文件。

read list
write list
这两个参数分别指定只允许对一个服务进行只读访问或同时有"写"权限的主机列表，这里，如果一个主机同时在两个列表里，则"写"优先。

704 ☆通过浏览限制

browsable
browseable
这两个功能相同的参数控制该段是否在浏览列表中列出。浏览列表列出的是可获得的共享列表。一般[home]段禁止浏览，以提高安全性。

705 ☆testparm

这个工具可让您测试 smb.conf 定义是否正确。

706 ☆testprns

这个工具可让您测试定义在 printcap 档里的打印机。

707 ☆swat

这个工具让您可使用 web 接口(如:IE、Netscape)对 Samba Server 做设定。

708 ☆nmblookup

这个工具可使用 NT/2000 网域内的主机名称查询出对应的 IP 地址。

nmblookup 命令用于把一个 NetBIOS 名字映射到 IP 地址。
/usr/bin/nmblookup sale

nmblookup *

709☆smbstatus

smbstatus 指令用于显示当前连接到的 SMB 服务器的 client 端连接状态。
[root@panda samba]# smbstatus

Samba version 3.0.10-1.4E
PID Username Group Machine

Service pid machine Connected at

No locked files

710☆smbpasswd

这个工具可用来建立、变更登入到 Samba server 的加密密码。

smbpasswd -a Linux 帐户名
注意:使用 smbpasswd -a username 之前，请先确定 /etc/passwd 档里存在 username 帐号。
smbpasswd -d username:停用 username 帐号
smbpasswd -e username:启用 username 帐号

要想让 Samba 使用与 Windows NT 兼容的口令加密算法,则必须在 Samba 服务器上保存包含与用户名对应的 hashed 值的口令文件。可以使用/usr/bin/mksmbpasswd.sh 程序产生加密口令文件的框架，文件名由"smb passwd file"参数设置，这里是/etc/samba/smbpasswd，命令行格式为:
cat /etc/passwd |mksmbpasswd.sh >/etc/samba/smbpasswd
然后 root 可以使用 smbpasswd 命令可以设置用户口令，
/usr/bin/smbpasswd cuckoo
root 还可以使用 smbpasswd 进行添加用户、取消口令等操作。但添加的用户必须在/etc/passwd 文件中存在。普通用户使用 smbpasswd
命令只能像使用 passwd 程序一样更改自己的 smb 口令。

711☆smbclient

smbclient 是 Samba 的 Linux 客户端,利用这个工具可连接其它 Unix like 的 Samba Server，或是连接 Windows 机器，以取得文件分享服务。

smbclient 是访问 SMB 服务器资源的客户程序。该程序提供的接口与 ftp 程序类似，访问操作包括从 SMB 服务器下载文件到本地，或从本地上载文件到 SMB 服务器，还可以在 SMB 服务器上检索目录信息等。

命令语法如下:
/usr/bin/smbclient //smbserver/service [passwd][-U username]

其中，smbserver 是 SMB 服务器的 NetBIOS 名，一般与服务器的主机名一样，但不是必须的。
"service"为服务器提供的服务，如文件服务 public 或打印服务 printer。
passwd 是访问某些服务时需要的口令。如果在命令行输入，后面则不会再提示输入口令。否则，将在后面提示输入口令。虽然有些要访问的服务器不需要口令，但仍然会有输入口令的提示。如果不希望有口令提示，则可以在命令行使用"-N"选项。
如果要使用其他的端口与 SMB 服务器进行 TCP 连接，可以使用"-p"选项来指定，缺省值为 139。
"-U"可以指定与 SMB 服务器连接时使用的用户名。如果没有指定，smbclient 使用环境变量 USER 指定的值作用户名。如果没有 USER 环境变量，则用"guest"。

在环境变量 USER 中，可以设置"USER=username%password"，这样就不用在命令行上输入口令。同时，使用 ps 命令看不到命令行参数，具有一定的安全性。同样，也可以使用"-U username%password"的形式指定用户名和口令。"-U"指定的口令"%password"要优先于在命令行[password]中指定的口令。

"-L"选项可以列出在一个服务器上提供的服务
/usr/bin/smbclient -L smbserver

查看服务器上的资源;
smbclient -L //IP/路径 [-U 用户名] #没有路径得话,不能加//
Sharename Type Comment
--------- ---- -------
IPC$ IPC 远程 IPC
HPLaserJ Printer HP LaserJet 6P
ADMIN$ Disk 远程管理
littlep Disk
C$ Disk 默认共享
Server Comment
--------- -------
GLASS
GRIND
Workgroup Master
--------- -------
BLUESUN GLASS
WORKGROUP HEIHEI

第一段列举了该机器(glass)上面的共享资源，第二段列举了 glass 所在的工作组里面所有提供 samba 服务的机器，第三列举了其他工作组提供 browse 服务的 Master 机器

使用网络资源
smbclient //IP 地址或者 NETBIOS 名称/共享资源名 [-U 用户名]
使用下面的命令可以进入与 ftp 类似的提示状态:
/usr/bin/smbclient //smbserver/service
执行结果:
smb:\>
其中"\"表示服务器当前的工作目录。在该提示符下可以使用的命令可以通过"?"、"help"命令列出。命令不分大小写。
命令 说明
?或 help [command] 提供关于帮助或某个命令的帮助
![shell command] 执行所用的 SHELL 命令，或让用户进入 SHELL 提示符
cd [目录] 切换到服务器端的指定目录，如未指定，则 smbclient 返回当前本地目录
lcd [目录] 切换到客户端指定的目录;
dir 或 ls 列出当前目录下的文件;
exit 或 quit 退出 smbclient
get file1 file2 从服务器上下载 file1，并以文件名 file2 存在本地机上;如果不想改名，可以把 file2 省略
mget file1 file2 file3 filen 从服务器上下载多个文件;
md 或 mkdir 目录 在服务器上创建目录
rd 或 rmdir 目录 删除服务器上的目录
put file1 [file2] 向服务器上传一个文件 file1,传到服务器上改名为 file2;
mput file1 file2 filen 向服务器上传多个文件

在该提示符下除了可以使用很多与 ftp 类似的命令外，还有命令 recurse，它可以被设置为 on，以便在使用 mget 或 mput 命令时能够 get 或 put 匹配的子目录。recurse 缺省为 off，即只 get 或 put 当前目录中的文件。

在该提示符下还可以使用下面的命令:
smb:\>tar c localfile filename
将服务器上文件名为"filename"的文件和目录(可以使用通配符)经过 tar 拷贝到本地的"localfile "文件中。
或者:
smb:\>tar x localfile filename
可以把本地 tar 文件 localfile 中的文件 filename 拷贝到服务器上;如果 filename 省略，则拷贝所有的文件。
共享备份
对您的邻居的共享进行备份。通过用户的帐户建立一个共享的数据打包，使用或者 smbtar 命令或者 smbclient 的-T 选项。

上面的命令也可以在 smbclient 命令行上来执行，如:
/usr/bin/smbclient //smbserver/service [passwd] -T c localfile filename
/usr/bin/smbclient //smbserver/service [passwd] -T x localfile filename

712 ☆smbprint

smbprint 是在 Linux 系统下将打印文件提交给远地 SMB 服务器打印的脚本程序。程序中调用的命令就是 smbclient。下面举例说明此命令的使用方法。

远端 SMB 服务器 NetBios 名为 prtserver(Windows 9X)，提供打印服务。服务名为 netpcl(NEC 激光打印机)，不需要口令。

在 Linux 系统中配置如下:
编辑/etc/printcap 文件，加入下面一行
smb:lo=/dev/null:sd=/usr/spool/smb:sh:if=/usr/bin/smbprint:af=/var/spool/lpd/smb/acct

建立目录/usr/spool/smb、/var/spool/lpd/smb

用下面命令建立记帐文件
touch /var/spool/lpd/smb/acct

建立/var/spool/lpd/smb/.config 文件，包含下面内容:
server = prtserver
service = necpcl
password = ""

编辑/usr/bin/smbprint 文件，在 smbclient 命令参数中使"echo translate"有效，这样可以使 UNIX 文件到 Windows 9x 系统打印时，进行<LF>与<CR/LF>的转换。

启动 lpd:
/etc/rc.d/init.d/lpd start

使用 lpr 命令打印/etc/hosts 文件
/lpr -P smb /etc/hosts
lpr 缺省使用的打印机名为 lp，这里使用的是 smb。

713 ☆smbtar

smbtar is a very small shell script on top of smbclient(1) which dumps SMB shares directly to tape.

OPTIONS

-s server
The SMB/CIFS server that the share resides upon.

-x service
The share name on the server to connect to. The default is "backup".

-X Exclude mode. Exclude filenames... from tar create or restore.

-d directory
Change to initial directory before restoring / backing up files.

-v Verbose mode.

-p password
The password to use to access a share. Default: none

-u user
The user id to connect as. Default: UNIX login name.

-a Reset DOS archive bit mode to indicate file has been archived.

-t tape
Tape device. May be regular file or tape device. Default: $TAPE environmental variable; if not set, a file called tar.out .

-b blocksize
Blocking factor. Defaults to 20. Seetar(1) for a fuller explanation.

-N filename
Backup only files newer than filename. Could be used (for example) on a log file to implement incremental backups.

-i Incremental mode; tar files are only backed up if they have the archive bit set. The archive bit is reset after each file is read.

-r Restore. Files are restored to the share from the tar file.

-l log level
Log (debug) level. Corresponds to the-d flag of smbclient(1).

smbtar -s 服务器 -u 用户 -x 共享名 -t 输出

714 ☆Samba 做 WINS Server

Samba 最大的功用除了做 file Server 外，还可以做 WINS Server。
Wins Server 最大的好处是可在 NT/2000 网域内做名称解析

传统上在 Internet 上做 FQDN(Full Quality Domain Name)与 IP address 之间的转换是藉由 DNS 做解析
但在局域网络里尤其是 NT/2000 网域，当您使用 NT/2000 的 WINS Server 与 NT/2000 的 DNS 做配搭时，假设有一台主机名称叫做 panda，所属领域是 panda.com，IP 地址是 192.168.152.128，目前此部主机尚未在 DNS 上注册，但是在 DNS 上激活正反查都可藉由 WINS 解析，当您使用 nslookup 查询 panda 主机时，您会发现可解析出 192.168.0.229，并且观察 DNS 上正查区域内，DNS 自动将 panda 以 A 资源纪录指定 IP 192.168.152.128，而反查区域内，DNS 自动将 128.152.168.192.in-addr.arpa 指到 panda.panda.com，这就是 WINS 的好处。

以往在 NT/2000 网域内使用 DNS+WINS Server 可免去在 DNS 上登录主机记录，若再搭配 DHCP Server 那么连 Client 端主机的 IP 也免输入，一切由 DHCP Server 分配动态 IP，再由 DNS+WINS Server 自动登录主机 FQDN 与 IP 解析关系，这个简单、易用组织网域的方法，如今也可以由 DNS+Linux 下的 Samba+Linux 下的 DHCP 来取代，因为 Samba 可担任 WINS Server 或指定 WINS Server 做网域名称解析。假设目前欲做 WINS+DHCP 的 Linux 主机 IP 为 192.168.0.229，其设定方式如下:

首先确定已激活 DNS Server，并且在 DNS 里启用正、反查皆可透过 WINS 做解析。

以下设定 Samba 为 Wins Server
wins support = yes

设定完成后，可使用 nmblookup 查询名称解析，假设目前有一主机其名称为 panda，使用动态分配 IP 联机，所配得的 IP 地址是 192.168.152.128，那么当您使用 nmblookup panda 做名称查询时，则出现下列画面:

[root@panda samba]# nmblookup panda
querying panda on 192.168.152.255
192.168.152.1 panda<00>
192.168.192.1 panda<00>

715 ☆Samba 做 File server

这种方式要确定你的内核支持 smbfs，如果没有支持的话，编译内核的时候必须选中 File systems--->Network File Systems---> SMB file system support (to mount Windows shares etc.)

smbmount //IP 或者 NETBIOS 名称/共享资源名 /本地挂接点 [-o option]

常用的的 option 有 username=<用户名>，password=<密码>，guest(指定为用 guest 访问，不用提供密码，前面的即使用 username=guest 参数的话也会要求输入密码)，ro(有时候为了系统安全要指定为只读模式)，rw，同时多个 option 的话用逗号隔开。

smbmount 的用法:
smbmount -o username=用户名,password=密码 //ip 地址或计算机名/共享文件夹名 挂载点
smbmount //ip 地址或计算机名/共享文件夹名 挂载点

或者可以用 mount -t smbfs [-o option] //IP 或者 NETBIOS 名称/共享资源名 /本地挂接点
实现同样的功能。

mount 挂载 smbfs 的用法;
mount -t smbfs -o codepage=cp936,username=用户名,password=密码 //ip 地址/共享文件夹名 挂载点
或
mount -t smbfs -o codepage=cp936,username=用户名,password=密码 //计算机名/共享文件夹名 挂载点
或
mount -t smbfs -o codepage=cp936 //ip 地址或计算机名/共享文件夹名 挂载点

不需要输入密码
smbmount //glass/littlep /test -o guest
或者
mount -t smbfs -o guest //glass/littlep /test
然后就能通过访问/test 来使用网络上的资源了。

如果不需要使用的时候，可以简单地使用 smbumount /test 或者 umount /test 来解除这个挂接。

在 mount 的命令中，我们发现有这样的一个参数 codepage=cp936，这是服务器端文件系统的编码的指定，cp936 就是简体中文，当然您可以用 utf8 等，尝试一下吧。

716 ☆Samba 做 Printer Server

system-config-printer 建立一个新的打印队列。
把打印队列命名为 printerX(其中 X 为您的工作站的号码)。配置打印机到本地连接的打印机 /dev/lp0。配置打印队列确保任何递交的打印作业将保留在队列中。不要忘记重新启动 samba 服务器。

setup 设置打印机

格式如下：
smb://用户:密码@主机名/共享打印机名

例如我的设置：
smb://joo:ca0zh0ng@red/Legend LJ 880

smbclient //localhost/printerX -u panda
注意是队列名

/etc/printcap
lptest|lptest:rm=panda:rp=lptest:

717 ☆Windows 中访问 Samba 服务器共享

这个简单吧，在网上领居，查看工作组就能看得到，或者在浏览器上输入如下的
\\ip 地址或计算机名

718 ☆Samba 的 Client 端

若设定使用「Linux 主机的帐号与密码做为登入 Samba Server 的帐号与密码」时，由于 win95OSR2 之前版本及 NT SP3 之前版本的密码机制是不加密的纯文字密码，当登入 Samba Server 时可传送不加密的纯文字密码给 Samba，因而轻易认证登入。

而 win98 及 NT SP3 之后版本及 WIn2000 的密码机制是属于加过密的密码，当登入 Samba Server 时所传送的是加密的密码给 Samba，因而无法认证登入，此时，必须为需要登入 Samba Server 的使用者帐号，利用 smbpasswd 这个工具程序建立登入 Samba 的密码，当 windows Client 端登入 Samba 时，就可接受加过密的密码做为认证，不过此时就无需使用「Linux 主机的帐号与密码做为登入 Samba Server 的帐号与密码」，因为已经为使用者建立属于登入 Samba 的密码。必须注意的是:登入 Samba 的使用者帐号必须是已存在于 /etc/passwd 文件内的 Linux 系统帐号，唯有如此，才能确保能使用个人帐号目录。

不加密的联机

由于 Samba 内定的接受密码机制属于纯文字密码，当设定好 Samba Server 端后，此时若 Client 端是 Windows 95 OSR2 之前的版本或是 Win NT SP3 之前的版本，在登入本机时(假设使用 abc 帐号)，直接使用『网上邻居』即可看「home」、「abc」、「works」三个分享目录，这是因为 Client 传送纯文字密码到 Server 端做为认证。

若 Client 端是 Windows 95 OSR2 以后的版本或是 Win NT SP3 以后或是 Win2000 的版本，因为 Client 传送加密码到 Samba Server 端做为认证，则需设定为纯文字密码才可以与 Samba Server 联机。您必须使用『登入记录编辑器』(regedit)对联机密码形式做设定:

Windows 95/98 ==> samba_9x.reg
REGEDIT4
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]

"EnablePlainTextPassword"=dword:00000001

Windows NT ==> samba_nt.reg
REGEDIT4
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdr\Parameters]
"EnablePlainTextPassword"=dword:00000001

Windows 2000 ==>samba_2000.reg
REGEDIT4
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkStation\Parameters]
"EnablePlainTextPassword"=dword:00000001

你可以手动用 regedit 改,也可以存成 *.reg 直接执行!!

使用纯文字密码登入 Samba 的好处是:可享用现成的 Linux 帐号与密码，但却失去了安全性，此时可采用加密的密码登入 Samba。为了让 Samba 能对加密的密码做认证，除了在基本设定里需选择「Encrypted password required」外，还需要使用 smbpasswd 为每一帐号建立 samba 密码，其步骤如下:
vi /etc/samba/smb.conf
[global]项目设定加密，如下所示:
encrypt passwords = yes

719 ☆Samba 建立 PDC 域服务器

其实早在 samba2.2 版本已经能非常好的支持 samba 做 PDC(主域控制器)，只不过到了 3.0 对域的支持更加好，到现在为止最新的版本 3.0,已经支持 AD，并且支持 Microsoft Kerberos 认证、完全重写和可配置的认证子系统等新功能。

修改/etc/samba/smb.conf

[global]
workgroup = panda
netbios name = proxy
server string = Samba PDC running %v
socket options = TCP_NODELAY IPTOS_LOWDELAY SO_SNDBUF=8192 SO_RCVBUF=8192
这里的 workgroup = panda 就代表 panda 域，当然如果用 panda.com 那就更加规范，但为了客户端输入的方便，还是直接 panda 的好，netbios name = proxy 表示这台服务器的 netbios 名,socket options 选项设置控制 TCP/IP 性能。所显示的设置就可以与基于 Linux 的系统一起很好地工作了。

os level = 64
preferred master = yes

local master = yes
domain master = yes
#domain master 选项是一个"开关"，通告 Samba 将成为主域控制器。(local master browser)是维护局域网机器列表的服务器被称为本地主浏览器。

security = user
encrypt passwords = yes
domain logons = yes
log file = /var/log/samba/log.%m
log level = 2
max log size = 50
hosts allow = 127.0.0.1 192.168.1.0/255.255.255.0
#这里我们还是使用 user 验证方式，不要悬在所谓的 domain，至于 hosts allow 大家可以根据自己的需求写那些网段可以访问你的服务器，或者索性不写也行。

logon home = \\%L\%U\.profile
logon drive = H:
logon path = \\%L\profiles\%U
logon script = netlogon.bat
#以上是漫游设置和登录脚本，logon path = \\%L\profiles\%U，会于下面我们要说的 [profiles] 共享成对应关系。

[homes]
comment = Home Directories
browseable = no
writeable = yes

[profiles]
path = /home/samba/profiles
writeable = yes
browseable = no
create mask = 0600
directory mask = 0700

netlogon 共享是登录到域服务器时必须有的共享目录，其 netlogon 共享目录设置如下:
[netlogon]
comment = Network Logon Service
path = /home/netlogon
read only = yes
browseable = no
write list= root

以上是关于共享的手腕子，其中 profile 是用来存放每个登录用户的设置文件，以便用户以后登录可以从服务器读取以前的桌面设置，netlogon 是用来存放登录脚本的，所以要限制写的权限，假设这里只有 root 用户可以有权限。

将用户和机器帐户添加到域控制器。

先建立创建了下列各组以及创建两个必要目录，并设置正确的所有权。
groupadd admin
groupadd machines
mkdir -m 0775 /home/netlogon
chown root.admins /home/netlogon
mkdir /home/samba /home/samba/profiles
chown 1757 /home/samba/profiles

手动在 Samba 服务器上先创建机器帐号
useradd -g machines -d /dev/null -c "machine id" -s /bin/false ibm240$
passwd -l ibm240$
输入两遍密码;
不要忘记标上美元符号;这是必需的，它将该项标识为信任帐户

创建 linux 帐户后，我们现在可以将该机器添加到 /etc/samba/smbpasswd
smbpasswd -a -m ibm240

自动添加机器帐号
自动添加只要在[global]添加
add user script = /usr/sbin/useradd -d /dev/null -g machines -s /bin/false -M %u

添加用户帐号
首先添加的是 root 帐户，把 root 加入到 smb 帐户中
smbpasswd -c root
加入到 NT 域中需要管理员的干预所以我们还得将 ROOT 账号添加到 SAMBA 用户当中
这一步很重要，因为后面的加入域要用到有管理员的帐号加入域的权限，否则用普通用户好像不能顺利加入域

然后添加 SAMBA 普通用户
useradd frank
passwd frank
smbpasswd -a frank

为了方便以后的管理，最好 smb 的用户密码和 unix 系统密码一样，这样我们还可以用到 samba 的密码同步功能

#下面的选项语句将允许用户从 Windows 客户机上更改他们的 Samba 密码，这样会随即更新他们的 UNIX 密码以与新的 Samba 项相匹配。但是如果更改了 UNIX 密码，那么同一技术不能逆向工作;必需手工同步更改 Samba 密码。也是在[global]，初学者可以先不做这个工作。
unix password sync = yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*UNIX*password* %n\n *Retype*new*UNIX*password* %n\n *Enter*new*UNIX*password* %n\n *Retype*new*UNIX*password* %n\n *passwd: *all*authentication*tokens*updated*successfully*
#上述语句中唯一值得一提的是 passwd chat 选项，不管这里如何显示它，都要将它输入成一行。还要注意有些选项使用"password"，而有些使用"passwd"。

在新建用户的主目录下建立一个.profile.pds 文件夹，其作用是用来存放用户的配置文件。建立这个文件的时候最好是用 SU 切换到该用户(su test1)，这样建立的好处是直接拥有该文件夹的所有权，以便在用户登录的时候可以很好建立和存储配置文件
客户端的设置，这里由于条件的限制，我只试验了 windows2000 客户端加入域，至于 winxp 和 win98 的加入大家自己去试验。

Samba PDC 的配置就这样完成了。剩下唯一要做的是将客户机加入到域中。记得重启 samba 服务哦！

(win200 机器最好先重启一下，可以避免一些不必要的问题)然后转至 控制面板 -> 网络-> 网络标示，如果机器目前被配置在 工作组 选项下，那么选中 域 单选按钮并输入域名 panda。

现在，通过使用用户名 root 和相应的密码登录到域。必需初始化服务器和客户机机器之间的"秘密"。从此时起，任何已认证的用户都可以从这台机器登录。
应该出现一个欢迎您来到 XX 域的消息

720 ☆Samba 常见故障排除

1 SMB 服务器上的操作
1).使用下面的命令来查看是否有错误的配置。
/usr/bin/testparm /etc/smb.conf
2). 在服务器上以及客户机上用 ping 命令检查 tcp/ip 是否正常工作。
3). 在 SMB 服务器上用
/usr/bin/smbclient -L SMBserver
命令将得到一个共享的列表。
如果出现失败信息，则要检查与服务器相关的"hosts allow"、"hosts deny"、"valid users"、"invlid users"等参数的设置。如果出现"connection refused"，则检查 Samba 服务器进程是否启动;若 Samba 以守护进程形式运行的话用"netstat -a"命令检查端口是否处于监听状态。
4). 运行了命令"/usr/bin/nmblookup SMBserver"，将返回 SMB 服务器的 IP 地址。否则 nmbd 没有正确的运行。

5). 运行"/usr/bin/nmblookup -d 2 '*' "命令，在子网中运行 NetBIOS/TCPIP 的主机将会响应。否则说明 nmblookup 不能正确的得到广播地址，可以尝试在 smb.conf 文件中用 interfaces 参数人工设置 IP 地址、广播地址和子网掩码。
6). 运行"/usr/bin/smbclient \\SMBserver\homes"命令，需要输入客户在 SMB 服务器上的用户口令，然后会出现类似"smb>"的提示符。这时可以使用 dir 命令浏览客户在 SMB 服务器帐户目录下的文件，用 help 命令可以得到其他命令的帮助。如果不能出现类似"smb>"的提示符，并且有"invalid network name"的错误信息，则有可能 homes 共享没有正确的设置;或者有"Bad password"，则要检查"shadow password"、"password encryption"以及参数"valid users"和"path"的设置。

2、SMB 客户机上的操作
1). 运行"net view \\SMBserver"命令，应该能够列出 SMB 服务器上的共享。
如果出现类似"network name not found"的错误，则要检查客户机上 DNS 或 WINS 的设置。如果出现"Invalid network name"或"bad password error"的错误，则参照与上面"smbclient -L"命令中同样的错误结果的解决方法。要注意客户机将要用客户注册客户机时所用的用户名/口令去与 SMB 服务器连接，所以在 SMB 服务器上，用户最好有同样的用户名/口令。
2). 使用"net use x;\\SMBserver\homes"命令，可以把 SMB 服务器上客户 home 目录映射到客户机"x:"盘，"x:"应该是一个客户机上未用的盘号。这样，共享的目录可以在客户机上以"x:"盘来看待。正常的信息为"command complete successfully"。
如果要停止使用"x:"盘，可以使用"net use x:/delete"命令。如果不能正常的把客户的 home 目录映射到"x:"盘，则要检查 SMB 服务器上"hosts allow"等参数的配置。如果客户机上的用户名不能与服务器上的用户名匹配，尝试使用"username map"选项。
3). 如果在 SMB 服务器上没有使用"encrypted passwords"，则在客户机上要在注册表中允许使?quot;普通文本"格式的口令。

721☆samba 实验

添加 Samba 用户
在您的系统中增加几个用户，但是并不给他们设定密码。这些用户仅能够从 samba 服务访问服务器。为了使得他们在 shadow 中不含有密码，这些用户的 shell 应该设定为/sbin/nologin

useradd -s /bin/false karl
useradd -s /bin/false joe
useradd -s /bin/false mary
useradd -s /bin/false jen

smbpasswd -a karl
smbpasswd -a joe
smbpasswd -a mary
smbpasswd -a jen

homes 目录

注意到第一个在/etc/samba/smb.conf 设定的共享[home]并没有指定路径。该共享被配置用来当用户连接并且认证通过以后共享用户的 home 目录。浏览一个或者两个用户的 home 目录。上传一个文件到 joe 的 home 目录。

访问目录
smbclient //127.0.0.1/homes -U karl

smbclient //localhost/myshare -U mary
您应该看到 smb:\>提示符

建立组
建立一个对于拥有 gid 为 30000 的用户叫做 legal 的新组并且使用 usermod 命令将这些用户加到组里去。

groupadd -g 30000 legal
usermod -G legal karl
usermod -G legal joe
usermod -G legal mary
usermod -G legal jen

建立公共目录
建立一个目录/home/depts/legal。对于这个目录设定拥有权限，使得在 legal 组中的用户可以在这个目录中添加 / 删除文件，然而其他的人不可以。并且设定 SGID 和粘滞位使得所有在这个目中建立的文件都拥有同 legal 组的权限并且组中其他的的人不能够删除该用户建立的文件。
chown :legal /home/depts/legal/
chmod 3770 /home/depts/legal
ll -d /home/depts/legal
drwxrws--T 2 root legal 4096 7 月 18 21:42 /home/depts/legal

mkdir -p /home/depts/legal
chgrp legal /home/depts/legal
chmod 3770 /home/depts/legal

建立共享
在/etc/samba/smb.conf 中建立一个 samba 共享叫做[legal]。只有 legal 组中的用户才能够访问该共享。并且确保在[legal]中存放的文件的被建立的许可权限为 0600。

[legal]
commnet = Legal's files
path = /home/depts/legal
public = no
write list = @legal

create mask =0660

service smb restart

722 ☆NFS

NFS 即网络文件系统(Network File System)，是使不同的计算机之间能通过网络进行文件共享的一种网络协议，多用于 UNIX/Linux 网络系统中。一台 NFS 服务器就如同一台文件服务器，只要将其文件系统共享出来，NFS 客户端就可以将它挂载到本地系统中，从而可以像使用本地文件系统中的文件一样使用那些远程文件系统中的文件。

NFS 最早是由 Sun 公司于 1984 年开发出来的，其目的就是让不同计算机、不同操作系统之间可以彼此共享文件。

虽然 NFS 可以在网络中进行文件共享，但是 NFS 协议本身并没有提供数据传输的功能，它必须借助于远程过程调用(RPC)协议来实现数据的传输.

RPC 定义了一种进程间通过网络进行交互通信的机制,它允许客户端进程通过网络向远程服务器上的服务进程请求服务,而不需要了解底层的通信协议细节.

可以将 NFS 服务端看成一个 RPC 服务器,NFS 客户端看成一个 RPC 客户端,这样 NFS 服务端和 NFS 客户端之间就可以通过 RPC 协议进行数据传输.

使用 NFS 服务,至少需要三个守护进程
(1)rpc.nfsd
基本的 NFS 守护进程,管理客户端是否能登入服务器
(2)rpc.mountd
RPC 的安装守护进程,管理 NFS 的文件系统.通过 rpc.nfsd 登录后,还必须通过文件使用权限的验证,rpc.mountd 会读取/etc/exports 来对比客户端的权限
(3)portmap
主要实进行端口映射工作.把服务对应的端口提供给客户端,使客户端通过端口请求服务.注意,potrmap 只用于 RPC,但它对于 NFS 服务是必不可少的.portmap 没有运行,NFS 客户端就无法查找 NFS 服务器中共享的目录

723 ☆NFS 安装

目前几乎所有的 Linux 发行版都默认安装了 NFS 服务
rpm -q nfs-utils portmap

724 ☆NFS 启动和停止

启动和停止 NFS 服务
为了使 NFS 服务器能正常工作，需要启动 portmap 和 nfs 这两个服务，并且 portmap 一定要先于 nfs 启动
service portmap restart
service nfs restart

725 ☆NFS 防火墙

编辑/etc/sysconfig/nfs （默认是没有这个文件的）加入
#LOCKD_TCPPORT=2049
#LOCKD_UDPPORT=2049
MOUNTD_PORT=815

然后开放端口 2049 和 111(portmap)

service nfslock status

726 ☆NFS 测试

要检查 NFS 服务是否正常运行，可使用 rpcinfo -p 命令。如果 NFS 服务运行正常，就可在该命令执行结果中看到关于 portmapper、nfs 和 mountd 等守护进程的条目。

727 ☆exports

只需在 NFS 的主配置文件/etc/exports 中进行设置，然后启动 NFS 服务即可。

exports 文件的格式
在 exports 文件中，可以定义 NFS 系统的输出目录(即共享目录)、访问权限和允许访问的主机等参数。该文件默认为空，没有配置输出任何共享目录，这是基于安全性的考虑，这样即使系统启动 NFS 服务也不会输出任何共享资源。

exprots 文件中每一行提供了一个共享目录的设置
<输出目录> [客户端(选项 1,选项 2,...)] [客户端 2(选项 1.选项…)]

其中，除输出目录是必选参数外，其他参数都是可选的。

值得注意的是，格式中的输出目录和客户端之间、客户端与客户端之间都使用空格分隔，但是客户端和选项之间不能有空格。

1.输出目录

输出目录是指 NFS 系统中需要共享给客户端使用的目录。

2.客户端
客户端是指网络中可以访问这个 NFS 输出目录的计算机。客户端的指定非常灵活，可以是单个主机的 IP 地址或域名，也可以是某个子网或域中的主机等。

192.168.16.20　单机
192.168.16.0/24　子网
pc1.panda.com　单机
*.panda.com 域
*(或缺省)　所有

3.选项
选项用来设置输出目录的访问权限、用户映射等。分三类

(1)访问权限选项
用于控制输出目录访问权限的选项。这类选项只有 ro 和 rw 两项
ro　设置输出目录只读
rw　设置输出目录可读写

(2)用户映射选项
在默认情况下，当客户端访问 NFS 服务器时，若远程访问的用户是 root 用户，则 NFS 服务器会将它映射成一个本地的匿名用户(该用户账户为 nfsnobody)，并将它所属的用户组也映射成匿名用户组(该用户组账户也为 nfsnobody)，这样有助于提高系统的安全性。用户映射选项可对此进行调整

all_squash　将远程访问的所有普通用户及所属用户组都映射为匿名用户或用户组(一般均为 nfsnobody)
no_all_squash　不将远程访问的所有普通用户及所属用户组都映射为匿名用户或用户组
root_squash　将 root 用户及所属用户组都映射为匿名用户或用户组(默认设置)
no_root_squash　不将 root 用户及所属用户组都映射为匿名用户或用户组
anonuid=xxx 将远程访问的所有用户都映射为匿名用户，并指定该匿名用户账户为本地用户账户(UID=xxx)
anongid=xxx 将远程访问的所有用户组都映射为匿名用户组账户，并指定该匿名用户组账户为本地用户组账户(GID=xxx)

(3)其他选项
其他选项比较多，可用于对输出目录进行更全面的控制
secure　限制客户端只能从小于 1024 的 TCP/IP 端口连接 NFS 服务器(默认设置)
insecure 允许客户端从大于 1024 的 TCP/IP 端口连接 NFS 服务器
sync将数据同步写入内存缓冲区与磁盘中，虽然这样做效率较低，但可以保证数据的一致性
async　将数据先保存在内存缓冲区中，必要时才写入磁盘

wdelay 检查是否有相关的写操作，如果有则将这些写操作一起执行，这样可提高效率(默认设置)
no_wdelay 若有写操作则立即执行，应与 sync 配合使用
subtree_check 若输出目录是一个子目录，则 NFS 服务器将检查其父目录的权限(默认设置)
no_subtree_check 即使输出目录是一个子目录，NFS 服务器也不检查其父目录的权限，这样做可提高效率

NFS 服务配置实例
/nfs/public 192.168.16.0/24(rw,async) *(ro)
/nfs/liu 192.168.16.20(rw,sync)
/nfs/root *.panda.com(ro,no_root_squash)
/nfs/users *.panda.com(rw,insecure,all_squash,sync,no_wdelay)
/mnt/cdrom 192.168.16.*(ro)
能否真正地写入，还要看该目录对该用户有没有开放 Linux 文件系统权限的写入权限。
如果该用户是普通用户，那么只有当该目录对该用户开放了写入权限时，该用户才可以在该共享目录下创建子目录及文件，且新建子目录及文件的所有者就是该用户(实际上应该是该用户的 UID)。
如果该用户是 root 用户，由于默认选项中有 root_squash,root 用户会被映射为 nfsnobody,因此只有该共享目录对 nfsnobody 开放了写入权限时，该用户才能在共享目录中创建子目录及文件，且所有者将变成 nfsnobody。

728☆etab

检查输出目录所使用的选项
在配置文件/etc/exports 中，即使在命令行中只设置了一两个选项，但在真正输出目录时，实际上还带有很多默认的选项。通过查看/var/lib/nfs/etab 文件，就可以了解到真正输出目录时，到底使用了什么选项

/var/lib/nfs/etab 文件中，可以看到有 anonuid=-2，anongid=-2，实际上应是 65536-2=65534，也就是说，匿名用户账户及所属用户组账户的 ID 分别为 UID=65534、GID=65534。对照/etc/passwd 和/etc/group 文件，可以知道它就是 nfsnobody。

729☆exportfs

维护 NFS 服务的输出目录列表
每当修改/etc/exports 文件的内容后，实际上不需要重新启动 NFS 服务，而直接使用命令 exportfs 就可以使设置立即生效。
exportfs 命令就是用来维护 NFS 服务的输出目录列表的
exportfs [选项]
-a:输出/etc/exports 文件中所设置的所有目录
-r:重新读取/etc/exports 文件中的设置，并使设置立即生效，而不需重新启动 NFS 服务
-u:停止输出某一目录
-v:在输出目录时将目录显示到屏幕上

1.重新输出共享目录
每当修改了/etc/exports 文件的内容后，可使用下面的命令来重新输出共享目录。
exportfs -rv

2.停止输出所有共享目录
要停止当前主机中 NFS 服务器的所有共享目录输出，可使用下面的命令
exportfs -auv

showmount -e
看当前主机输出的共享,此时为空

730 ☆showmount

使用 showmount 命令测试 NFS 服务器的输出目录状态
showmount [选项] NFS 服务器名称或地址
-a:显示指定的 NFS 服务器的所有客户端主机及其所连接的目录;
-d:显示指定的 NFS 服务器中已被客户端连接的所有输出目录;
-e:显示指定的 NFS 服务器上所有输出的共享目录。

(1)查看当前主机中 NFS 服务器上所有输出的共享目录
使用不带 NFS 服务器名称或地址参数的 showmount -e 命令，可以查看当前主机中 NFS 服务器上所有输出的共享目录

(2)显示当前主机中 NFS 服务器上被挂载的所有输出目录
使用不带 NFS 服务器名称或地址参数的 showmount -d 命令，可以查看当前主机中 NFS 服务器上所有被挂载的输出目录

731 ☆NFS 客户端

在 NFS 服务器设置完成后，客户端可以先查看 NFS 服务器上有哪些共享目录，然后使用 mount 命令将可用的共享目录挂载到本机的文件系统中，甚至还可以实现开机自动挂载，以后用户就可像使用本地文件系统中的目录一样使用 NFS 挂载目录。

查看 NFS 服务器信息
在客户端，要查看 NFS 服务器上有哪些共享目录，可以使用 showmount 命令
showmount -e 192.168.152.128

出现错误:(没有收集)
原因可能是 NFS 服务器上没有启动 portmap 或 nfs 服务，也可能是被防火墙过滤掉了。解决办法是启动 NFS 服务器上的 portmap 或 nfs 服务，并重新设置 NFS 服务器上的防火墙(包括 iptables 和

TCP-Wrappers)若是 iptables 防火墙引起的故障，为了测试 NFS 服务器的功能,可以简单的使用 services iptables stop 命令先关掉该防火墙。

连接 NFS 服务器
在利用 showmount 得知远程 NFS 服务器上的共享资源后，接下来就是进行实际的挂载操作
挂载 NFS 服务器上的输出目录的命令格式为
mount -t nfs 服务器名或 IP 地址:输出目录 本地挂载目录

mount -t nfs 192.168.152.128:/nfs/public /mnt/nfs
/nfs/public 是绝对目录,不是相对于 nfs 根的目录

值得注意的是，当客户端上的用户无权访问 NFS 服务器上的输出目录时，就会无法进行挂载。

卸载 NFS 服务器
如果不想再使用已挂载的 NFS 输出目录，可用 umount 命令来卸载该目录。
umount /mnt/nfs

如果当前还有客户端正在连接 NFS 服务器，此时要想将 NFS 服务器所在的主机关机，应先关掉 portmap 和 nfs 这两个服务，否则可能要等待很久才能正常关机。如果无法正确地将 potrmap 和 nfs 这两个服务关掉，那么可先用命令 netstat -utlp 找出它们的 PID，然后使用 kill 命令杀掉，这样才能正常关机。
当然，还可以先用 exportfs-auv 命令将当前主机中 NFS 服务器的所有输出目录停止掉。再关机。

启动时自动连接 NFS
要想让系统在启动时自动挂载 NFS 服务器上的输出目录，应编辑文件/etc/fstab，在该文件中加入如下格式的语句。
NFS 服务器名或 IP 地址:输出目录 本地挂裁目录 nfs defaults 0 0

192.168.152.128:/nfs/public /mnt/nfs nfs defaults 0 0

732 ☆FTP

FTP，file transfer protocol，这是文件传输的通讯协议，也是一般最常用来传送文件的方式。

FTP 分为两类，一种为 PORT FTP，也就是一般的 FTP;另一类是 PASV FTP，分述如下:

PORT FTP
这是一般形式的 FTP，首先会建立控制频道，默认值是 port 21，也就是跟 port 21 建立联机，并透过此联机下达指令。第二，由 FTP server 端会建立数据传输频道，默认值为 20，也就是跟 port 20 建立联机，并透过 port 20 作数据的传输。
在主动模式中，FTP 客户端随机开启一个大于 1024 的端口 x 向服务器的 2l 号端口发起控制连接请求后，开放 x+l 号端口进行监听；FTP 服务器接受请求并建立控制连接会话。如果客户端在控

制会话中发送数据连接请求，服务器在接收到命令后，会用其本地的 FTP 数据端口(通常是 20)来连接客户端指定的端口 X+1 进行数据传输

PASV FTP
跟 PORT FTP 类似，首先会建立控制频道，默认值是 port 21，也就是跟 port 21 建立联机，并透过此联机下达指令。第二，会由 client 端做出数据传输的请求，包括数据传输 port 的数字。
首先
FTP 客户端随机开启一个大于 1024 的端口 x 向服务器的 2l 端口发起连接，同时会开启 x+l 端口。然后向服务器发送 PASV 命令，通知服务器自己处于被动模式。服务器收到命令后，会开放一个大于 1024 的端口 Y 进行监听，然后用 PORT Y 命令通知客户端，自己的数据端口是 Y。客户端收到命令后，会通过 x+l 号端口连接服务器的端口 Y，然后在两个端口之间进行数据传输。

这两者的差异为何？PORT FTP 当中的数据传输 port 是由 FTP server 指定，而 PASV FTP 的数据传输 port 是由 FTP client 决定。

通常我们使用 PASV FTP，是在有防火墙的环境之下，透过 client 与 server 的沟通，决定数据传输的 port。

733☆ftp 用户管理

FTP 服务器对用户的管理，在默认的情况下是根据 /etc/passwd 及/etc/group 来进行的

如果 FTP 允许匿名登录的话;
ftp://mirrors.kernel.org
或
ftp://ftp:ftp@mirrors.kernel.org
都是一样的

如果我们以 ftp 命令连接 mirrors.kernel.org 时，我们会发现需要输入用户 ftp，密码 ftp 才能访问;

ftp 这个用户可以在您的操作系统中的 /etc/passwd 中能找得到;可能有类似下面的一行;
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

/var/ftp 是 ftp 用户的家目录，可以自己来定义;
/sbin/nologin 这是用户登录 SHELL ，这个也是可以定义的，/sbin/nologin 表示不能登录系统;系统虚拟帐号(也被称为伪用户)一般都是这么设置。
比如我们把 ftp 用户的/sbin/nologin 改为 /bin/bash ，这样 ftp 用户通过本地或者远程工具 ssh 或 telnet 以真实用户身份登录到系统。这样做对系统来说是不安全的;如果您认为一个用户没有太大的必要登录到系统，就可以只给他 FTP 帐号的权限，也就是说只给他 FTP 的权限，而不要把他的 SHELL 设置成 /bin/bash 等;

ftp 用户组

我们查看 /etc/group 的时候，会发现类似这样一条;
ftp:x:50:
/etc/group 是用户组的管理配置文件，上面这行表示用户组 ftp，x 是密码段，50 是 GID;我们对照在/etc/passwd 中的 ftp 那行得知 ftp 用户是属于 ftp 用户组的，因为 ftp 用户那行中的 GID 和 ftp 用户组的 GID 是相同的;

匿名 ftp 用户和 ftp 用户组是不可以删除
在一般情况下是不能把/etc/passwd 和/etc/group 中有把 ftp 用户和用户组的行删除的，因为 FTP 服务器是需要他们来对 FTP 用户实现管理，在默认的情况下。虽然不能删除，但对/etc/passwd 及/etc/group 中的 ftp 用户及 ftp 用户组的一些相关的东西是能修改的;比如我们可以把 ftp 用户的家目录改掉，也可以把 ftp 用户的 UID 改掉 ... ...

734 ☆vsFTPd 的安装

[root@localhost ~]# rpm -ivh vsftpd*.rpm

735 ☆vsFTPd 启动关闭

service vsftpd restart

vsFTPd 服务器是否运行起来;
[root@localhost ~]# pgrep vsftpd
4248

736 ☆vsFTPd 防火墙

在 Fedora/Redhat/CentOS 中，您要设置一下防火墙，可以把防火墙关掉，或者在自定义中让 ftp "通过"防火墙;
[root@localhost ~]# system-config-securitylevel-tui
或者
[root@panda ~]# iptables -I INPUT -p tcp --dport 21 -j ACCEPT
[root@panda ~]# iptables -I INPUT -p tcp --dport 20 -j ACCEPT

或者
禁止所有对内连接
iptables -P INPUT DROP
打开内对内连接
iptables -A INPUT -i lo -j ACCEPT
打开 FTP 命令端口
iptalbes -A INPUT -p tcp --dport 21 -j ACCEPT
打开客户端的已确认连接
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

打开连接 FTP 模块
modprobe ip_conntrack_ftp

或者
PORT FTP:
ftp port:2121(非默认的 21)
ftp data port:2020(非默认的 20)
执行以下两行指令，只允许 port 2121 以及 port 2020 开放，其它关闭。
iptables -A INPUT -p tcp -m multiport --dport 2121,2020 -j ACCEPT
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset

修改/etc/vsftpd/vsftpd.conf
新增底下两行
listen_port=2121
ftp_data_port=2020

PASV FTP:
ftp port:2121
ftp data port 从 9981 到 9986。
执行以下两行指令，只允许 port 2121 以及 port 9981-9990 开放，其它关闭。
iptables -A INPUT -p tcp -m multiport --dport
2121,9981,9982,9983,9984,9985,9986,9987,9988,9989,9990 -j ACCEPT
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset

修改/etc/vsftpd/vsftpd.conf
新增底下四行
listen_port=2121
pasv_enable=YES
pasv_min_port=9981
pasv_max_port=9986

737 ☆相关配置文件

/etc/vsftpd.ftpusers
这里面列出禁止登陆的用户名单

/etc/vsftpd.user_list
当配置文件中定义 userlist_deny=NO 时 这里面列出容许登陆的用户名单
当配置文件中定义 userlist_deny=YES 时这里面列出禁止登陆的用户名单(默认)

/etc/vsftpd/vsftpd.conf vsftpd 的基本配置文件

vsftpd 与 xinetd 相关的文件:
/etc/xinetd.conf xinetd 的基本配置文件
/etc/xinetd.d/vsftpd xinetd 引导 vsftpd 的配置文件(是否自动启动的选择也在这里)

执行文件在这个位置
/etc/sbin/vsftpd
/etc/sbin/xinetd

738☆vsftpd.conf

注意:修改完配置文件后，一定要重启 vsFTPd 服务器才能生效，切记~~
vsftpd.conf 配置文件就是 vsFTPd 服务器的全局控制文件，此配置文件中，每行应该算做一个规则;
前面带有#号的服务器不会解释，这和 apache 的配置文件类似;#后所接的内容一般是说明性的，或者是关掉某些功能的选项;
vsftpd.conf 的内容非常单纯，每一行即为一项设定。若是空白行或是开头为#的一行，将会被忽略。内容的格式只有一种，如下所示
option=value
要注意的是，等号两边不能加空白，不然是不正确的设定。
详细的帮助说明请参考 man vsftpd.conf

===ascii 设定==========
ascii_download_enable
管控是否可用 ASCII 模式下载。默认值为 NO。
ascii_upload_enable
管控是否可用 ASCII 模式上传。默认值为 NO。

===个别使用者设定==========
chroot_list_enable
用法:YES/NO
如果启动这项功能，则所有的本机使用者登入均可进到根目录之外的数据夹，除了列在 /etc/vsftpd.chroot_list 之中的使用者之外。默认值为 NO。
chroot_list_file=/etc/vsftpd.chroot_list
是否将系统用户限止在自己的 home 目录下,如果选择了 yes 那么
chroot_list_file=/etc/vsftpd.chroot_list 中列出的是 chroot 的用户的列表
建立 vsftp.chroot_list 文件，加入你的用户名，并将用户的 HOME 改成/var/ftp/，这样/var/ftp/就变成 /了
userlist_enable
用法:YES/NO
用 userlist_file 来限制用户访问.若是启动此功能，则会读取 userlist_file 当中的使用者名称。此项功能可以在询问密码前就出现失败讯息，而不需要检验密码的程序。默认值为关闭。注:开启 userlist_enable=yes 匿名帐号不能登陆
userlist_deny

用法:YES/NO
这个选项只有在 userlist_enable 启动时才会被检验。如果将这个选项设为 YES，则在 userlist_file 中的使用者将无法登入; 若设为 NO ， 则只有在 userlist_file 中的使用者才能登入。而且此项功能可以在询问密码前就出现错误讯息，而不需要检验密码的程序。
userlist_file
/指定文件存放的路径/ (文件放置的路径).缺省/etc/vsftpd.user_list
user_config_dir
定义个别使用者设定文件所在的目录，例如定义 user_config_dir=/etc/vsftpd/userconf，且主机上有使用者 test1,test2，那我们可以在 user_config_dir 的目录新增文件名为 test1 以及 test2。若是 test1 登入，则会读取 user_config_dir 下的 test1 这个文件内的设定。默认值为无。

＝＝＝欢迎语设定＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝
dirmessage_enable
用法:YES/NO
如果启动这个选项，使用者第一次进入一个目录时，会检查该目录下是否有.message 这个文件，若是有，则会出现此文件的内容，通常这个文件会放置欢迎话语，或是对该目录的说明。默认值为开启。
banner_file
当使用者登入时，会显示此设定所在的文件内容，通常为欢迎话语或是说明。默认值为无。
ftpd_banner
这边可定义欢迎话语的字符串，相较于 banner_file 是文件的形式，而 ftpd_banner 是字串的格式。预设为无。
message_file
#设置访问一个目录时获得的目录信息文件的文件名,默认是.message

＝＝＝特殊安全设定＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝
chroot_local_user
如果设定为 YES，那么所有的本机的使用者都不可以切换到根目录以外的数据夹。预设值为 NO。
hide_ids
如果启动这项功能，所有文件的拥有者与群组都为 ftp，也就是使用者登入使用 ls -al 之类的指令，所看到的文件拥有者跟群组均为 ftp。默认值为关闭。
ls_recurse_enable
若是启动此功能，则允许登入者使用 ls -R 这个指令。默认值为 NO。
write_enable
用法:YES/NO
这个选项可以控制 FTP 的指令是否允许更改 file system，譬如 STOR、DELE、RNFR、RNTO、MKD、RMD、APPE 以及 SITE。预设是关闭。
setproctitle_enable
用法:YES/NO
启动这项功能，vsftpd 会将所有联机的状况已不同的 process 呈现出来，换句话说，使用 ps -ef 这类的指令就可以看到联机的状态。默认值为关闭。
tcp_wrappers

用法:YES/NO
如果启动，则会将 vsftpd 与 tcp wrapper 结合，也就是可以在/etc/hosts.allow 与/etc/hosts.deny 中定义可联机或是拒绝的来源地址。
pam_service_name
这边定义 PAM 所使用的名称，预设为 vsftpd。
secure_chroot_dir
这个选项必须指定一个空的数据夹且任何登入者都不能有写入的权限，当 vsftpd 不需要 file system 的权限时，就会将使用者限制在此数据夹中。默认值为/usr/share/empty

===日志文件设定==========
xferlog_enable
用法:YES/NO
如果启动，上传与下载的信息将被完整纪录在底下 xferlog_file 所定义的文件中。预设为开启。
xferlog_file
这个选项可设定日志文件所在的位置，默认值为/var/log/vsftpd.log。
xferlog_std_format
如果启动，则日志文件将会写为 xferlog 的标准格式，如同 wu-ftpd 一般。默认值为关闭。

===空闲超时设定==========
accept_timeout
接受建立联机的超时设定，单位为秒。默认值为 60。
connect_timeout
响应 PORT 方式的数据联机的超时设定，单位为秒。默认值为 60。
data_connection_timeout
建立数据联机的超时设定。默认值为 300 秒。数据传输超时时间
idle_session_timeout
发呆的超时设定，若是超出这时间没有数据的传送或是指令的输入，则会强迫断线，单位为秒。默认值为 300。断开不活跃 session 的时间

===速率限制==========
anon_max_rate
匿名登入所能使用的最大传输速度，单位为每秒多少 bytes，0 表示不限速度。默认值为 0。
local_max_rate
本机使用者所能使用的最大传输速度，单位为每秒多少 bytes，0 表示不限速度。预设值为 0。

===新增文件权限设定==========
anon_umask
匿名登入者新增文件时的 umask 数值。默认值为 077。
file_open_mode
上传文件的权限，与 chmod 所使用的数值相同。默认值为 0666。
local_umask
本机登入者新增文件时的 umask 数值。默认值为 077。

===port 设定==========
connect_from_port_20
用法:YES/NO
若设为 YES，则强迫 ftp-data 的数据传送使用 port 20。默认值为 YES。
ftp_data_port
设定 ftp 数据联机所使用的 port。默认值为 20。
listen_port
FTP server 所使用的 port。默认值为 21。
pasv_max_port
建立资料联机所可以使用 port 范围的上界，0 表示任意。默认值为 0。
pasv_min_port
建立资料联机所可以使用 port 范围的下界，0 表示任意。默认值为 0。

===其它==========
anon_root
使用匿名登入时，所登入的目录。默认值为无。
local_enable
用法:YES/NO
启动此功能则允许本机使用者登入。默认值为 YES。
local_root
本机使用者登入时，将被更换到定义的目录下。默认值为无。
text_userdb_names
用法:YES/NO
当使用者登入后使用 ls -al 之类的指令查询该文件的管理权时，预设会出现拥有者的 UID，而不是该文件拥有者的名称。若是希望出现拥有者的名称，则将此功能开启。默认值为 NO。
pasv_enable
若是设为 NO，则不允许使用 PASV 的模式建立数据的联机。默认值为开启。
nopriv_user=ftpsecure
运行 vsftpd 需要的非特权系统用户默认是 nobody,连接到到 ftp 的用户都是 ftpsecure
async_abor_enable=YES
#是否允许运行特殊的 ftp 命令 async ABOR.

===更换文件所有权==========
chown_uploads
用法:YES/NO
若是启动，所有匿名上传数据的拥有者将被更换为 chown_username 当中所设定的使用者。这样的选项对于安全及管理，是很有用的。默认值为 NO。
chown_username
这里可以定义当匿名登入者上传文件时，该文件的拥有者将被置换的使用者名称。预设值为 root。

===guest 设定==========

guest_enable
用法:YES/NO
若是启动这项功能，所有的非匿名登入者都视为 guest。默认值为关闭。
guest_username
这里将定义 guest 的使用者名称。默认值为 ftp。

＝＝＝anonymous 设定＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝
anonymous_enable
用法:YES/NO
管控使否允许匿名登入，YES 为允许匿名登入，NO 为不允许。默认值为 YES。
no_anon_password
若是启动这项功能，则使用匿名登入时，不会询问密码。默认值为 NO。
anon_mkdir_write_enable
用法:YES/NO
如果设为 YES，匿名登入者会被允许新增目录，当然，匿名使用者必须要有对上层目录的写入权。默认值为 NO。
anon_other_write_enable
用法:YES/NO
如果设为 YES，匿名登入者会被允许更多于上传与建立目录之外的权限，譬如删除或是更名。默认值为 NO。
anon_upload_enable
用法:YES/NO
如果设为 YES，匿名登入者会被允许上传目录的权限，当然，匿名使用者必须要有对上层目录的写入权。默认值为 NO。
anon_world_readable_only
用法:YES/NO
如果设为 YES，匿名登入者会被允许下载可阅读的文件。默认值为 YES。
注:要注意文件夹的属性，匿名帐户是其它(other)用户要开启它的读写执行的权限
(R)读-----下传 (W)写----上传 (X)执行----如果不开 FTP 的目录都进不去
ftp_username
定义匿名登入的使用者名称。默认值为 ftp。
deny_email_enable
若是启动这项功能，则必须提供一个文件/etc/vsftpd.banner_emails，内容为 email address。若是使用匿名登入，则会要求输入 email address，若输入的 email address 在此文件内，则不允许联机。默认值为 NO。
banned_email_file=/etc/vsftpd.banned_emails
是否允许禁止匿名用户使用某些邮件地址,如果是输入禁止的邮件地址的路径和文件名

＝＝＝Standalone 选项＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝
listen
用法:YES/NO

若是启动，则 vsftpd 将会以独立运作的方式执行，若是 vsftpd 独立执行，如 RedHat9 的默认值，则必须启动;若是 vsftpd 包含在 xinetd 之中，则必须关闭此功能，如 RedHat8。在 RedHat9 的默认值为 YES。
listen_address
若是 vsftpd 使用 standalone 的模式，可使用这个参数定义使用哪个 IP address 提供这项服务，若是主机上只有定义一个 IP address，则此选项不需使用，若是有多个 IP address，可定义在哪个 IP address 上提供 ftp 服务。若是不设定，则所有的 IP address 均会提供此服务。默认值为无。
max_clients
若是 vsftpd 使用 standalone 的模式，可使用这个参数定义最大的总联机数。超过这个数目将会拒绝联机，0 表示不限。默认值为 0。
max_per_ip
若是 vsftpd 使用 standalone 的模式，可使用这个参数定义每个 ip address 所可以联机的数目。超过这个数目将会拒绝联机，0 表示不限。默认值为 0。

==========
总结
Anonymous_enable=yes (允许匿名登陆)
Dirmessage_enable=yes (切换目录时，显示目录下.message 的内容)
Local_umask=022 (FTP 上本地的文件权限，默认是 077)
Connect_form_port_20=yes (启用 FTP 数据端口的数据连接)*
Xferlog_enable=yes (激活上传和下传的日志)
Xferlog_std_format=yes (使用标准的日志格式)
Ftpd_banner=XXXXX (欢迎信息)
Pam_service_name=vsftpd (验证方式)*
Listen=yes (独立的 VSFTPD 服务器)*
Anon_upload_enable=yes (开放上传权限)
Anon_mkdir_write_enable=yes (可创建目录的同时可以在此目录中上传文件)
Write_enable=yes (开放本地用户写的权限)
Anon_other_write_enable=yes (匿名帐号可以有删除的权限)
Anon_world_readable_only=no (放开匿名用户浏览权限)
Ascii_upload_enable=yes (启用上传的 ASCII 传输方式)
Ascii_download_enable=yes (启用下载的 ASCII 传输方式)
Banner_file=/var/vsftpd_banner_file (用户连接后欢迎信息使用的是此文件中的相关信息)
Idle_session_timeout=600(秒) (用户会话空闲后 10 分钟)
Data_connection_timeout=120(秒)(将数据连接空闲 2 分钟断)
Accept_timeout=60(秒)(将客户端空闲 1 分钟后断)
Connect_timeout=60(秒)(中断 1 分钟后又重新连接)
Local_max_rate=50000(bite)(本地用户传输率 50K)
Anon_max_rate=30000(bite)(匿名用户传输率 30K)
Pasv_min_port=50000 (将客户端的数据连接端口改在
Pasv_max_port=60000 50000-60000 之间)
Max_clients=200 (FTP 的最大连接数)

Max_per_ip=4 (每 IP 的最大连接数)
Listen_port=5555 (从 5555 端口进行数据连接)
Local_enble=yes (本地帐户能够登陆)
Write_enable=no (本地帐户登陆后无权删除和修改文件)
这是一组
Chroot_local_user=yes (本地所有帐户都只能在自家目录)
Chroot_list_enable=yes (文件中的名单可以调用)
Chroot_list_file=/任意指定的路径/vsftpd.chroot_list
(前提是 chroot_local_user=no)
这又是一组
Userlist_enable=yes (在指定的文件中的用户不可以访问)
Userlist_deny=yes
Userlist_file=/指定的路径/vsftpd.user_list
又开始单的了
Banner_fail=/路径/文件名 (连接失败时显示文件中的内容)
Ls_recurse_enable=no
Async_abor_enable=yes
One_process_model=yes
Listen_address=10.2.2.2 (将虚拟服务绑定到某端口)
Guest_enable=yes (虚拟用户可以登陆)
Guest_username=所设的用户名 (将虚拟用户映射为本地用户)
User_config_dir=/任意指定的路径/为用户策略自己所建的文件夹
(指定不同虚拟用户配置文件的路径)
又是一组
Chown_uploads=yes (改变上传文件的所有者为 root)
Chown_username=root
又是一组
Deny_email_enable=yes (是否允许禁止匿名用户使用某些邮件地址)
Banned_email_file=//任意指定的路径/xx/
又是一组
Pasv_enable=yes (服务器端用被动模式)
User_config_dir=/任意指定的路径//任意文件目录 (指定虚拟用户存放配置文件的路径)

vsftpd.conf - vsftpd 的配置文件
描述
vsftpd.conf 可以用于控制 vsftpd, 以实现各种各样的功能. vsftpd 缺省到 /etc/vsftpd.conf 处查找此文件. 当然, 您也可以通过命令行参数进行指定. 这个命令行参数就是指 vsftpd 的配置文件. 对于想使用高级 inetd 管理的用户, 例如, xinetd, 则这个功能非常有用. 可以使用不同的配置文件来启动基于虚拟主机的每个服务.

格式

vsftpd.conf 的格式非常简单. 每行要么是注释, 要么是指令. 注释行以 # 开始, 将被忽略. 指令行格式如下:
选项=值
应当注意的一点是如果在 选项, = 和 值 之间存在空格, 将会报错.(译者注: 即三者之间不允许存在空格)
每项设定都有默认值, 这可以通过配置文件来修改.

布尔选项
下边是布尔选项的列表. 一个布尔选项的值可以被设为 YES 或 NO

allow_anon_ssl
只有在 ssl_enable 被激活时才有用. 如果设为 YES, 匿名用户将被允许使用安全的 SSL 联接.

默认: NO
anon_mkdir_write_enable
如果设为 YES, 匿名用户将允许在某些情况下创建目录. 这需要激活 write_enable 选项, 并且匿名 ftp 用户需要对父目录有写权限.

默认: NO
anon_other_write_enable
如果设为 YES, 匿名用户将拥有除 上载, 和创建目录 外更多的权限, 比如 删除和重命名. 通常不建议这么做, 但完整的配置文件是包括这一选项的.

默认: NO
anon_upload_enable
如果设为 YES, 匿名用户在某些情况下允许上载文件. 这需要将 write_enable 选项激活, 并且匿名用户应当对对应目录有写权限.

默认: NO
anon_world_readable_only
启用时, 将只允许匿名用户下载具有全球读权限的文件. 这将意味着 ftp 用户可以拥有自己的文件, 特别是前边提到的上载的文件.

默认: YES
anonymous_enable
用于控制是否允许匿名用户登录. 如果激活, ftp 和 anonymous 都将被视为匿名用户登录.

默认: YES
ascii_download_enable
如果被激活, 下载时将使用 ASCII 模式进行数据传输.

默认: NO

ascii_upload_enable

如果被激活, 上载时将使用 ASCII 模式进行数据传输.

默认: NO

async_abor_enable

如果被激活, 一个特别的 FTP 命令 "async ABOR" 将被激活. 只有某些 FTP 客户端需要使用这一特性. 另外, 这个特性并不是很好控制, 因此默认没有启用. 不幸的是, 如果没有启用这个特性, 某些 FTP 客户端在取消一个传输时就会挂起, 因此, 您可能希望启用它.

默认: NO

background

如果被激活, 并且 vsftpd 以 "listen" 模式启动, vsftpd 将会 background 监听进程. 即 control will immediately be returned to the shell which launched vsftpd.

默认: NO

check_shell

注意! 这个选项只对构建时加入 non-PAM 参数的 vsftpd 有效. 如果令其失效, vsftpd 将不会检查有效用户的用于本地登录的 /etc/shells.

默认: YES

chmod_enable

如果被激活, 将允许使用 SITE CHMOD 命令. 注意! 这只对本地用户有效. 匿名用户从不允许使用 SITE CHMOD.

默认: YES

chown_uploads

如果被激活, 所有匿名上载的文件的宿主将会调整为 chown_username 中指定的用户. 这样就便于管理, 特别是从安全的角度考虑.

默认: NO

chroot_list_enable

如果被激活, 您需要提供一个需要将其限制于其家目录中的本地用户列表. 如果将 chroot_local_user 设为 YES 则意义稍有不同. 在此情况下, 此列表变成不需将用户限制于其家目录的用户的列表. 默认情况下,这个列表文件是 /etc/vsftpd.chroot_list, 但可以通过 chroot_list_file 选项来设定.

默认: NO

chroot_local_user

如果设为 YES, 本地用户, 在登录后将(默认)被限制在其家目录中. 警告: 此选项有安全隐患, 特别是在用户拥有上载权限, 或可以 shell 访问的时候. 如果您不清楚后果, 请不要启用它. 注意,

这些安全隐患并不是 vsftpd 所特有的. 所有的提供将本地用户进行目录限制的 FTP 守护进程有存在这种隐患.

默认: NO

connect_from_port_20

用于控制在服务器端, 是否使用端口 20(ftp-data)进行数据联接. 基于安全的考虑, 有些客户端需要这样做. 相反, 禁用这个选项, 可以使 vsftpd 以较少特权运行.

默认: NO(但是在示例设置中启用了这个选项)

deny_email_enable

如果激活, 您应当提供一个禁止匿名用户用做密码的 e-mail 地址列表. 默认情况下, 这个列表文件为 /etc/vsftpd.banned_emails, 当然, 您可以通过 banned_email_file 选项指定.

默认: NO

dirlist_enable

如果设为 NO, 所有的目录列取命令都将被禁止.

默认: YES

dirmessage_enable

如果启用, 当用户首次进入一个新目录时, FTP 服务器将会显示欢迎信息. 默认情况下, 是扫描目录下的 .message 文件获取的, 当然, 您也可以通过 message_file 选项设定.

默认: NO(但是在示例设置中启用了这个选项)

download_enable

如果设为 NO, 所有的下载请求都将被拒绝.

默认: YES

dual_log_enable

如果启用, 将生成两个相似的日志文件, 默认在 /var/log/xferlog 和 /var/log/vsftpd.log 目录下. 前者是 wu-ftpd 类型的传输日志, 可以用于 标准工具分析. 后者是 vsftpd 自己类型的日志.

默认: NO

force_dot_files

如果激活, 以 . 开始的文件和目录在目录列取的时候将会被显示, 即使客户端没有使用 "a" 标识. 这不包括 "." 和 ".." 目录

默认: NO

force_local_data_ssl

只有在 ssl_enable 被激活时才能使用. 如果被激活, 则所有的 非匿名用户 登录时都被强制使用安全 SSL 联接来传送接收数据.

默认: YES

force_local_logins_ssl

只有在 ssl_enable 被激活时才能使用. 如果被激活, 则所有的 非匿名用户 登录时都被强制使用安全 SSL 联接来传送密码.

默认: YES

guest_enable

如果启用, 所有非匿名用户都将以 "guest" 身份登录. guest 通过 guest_username 设定, 来映射到一个指定用户.

默认: NO

hide_ids

如果启用, 所有目录中的用户和组信息列取时都将显示为 "ftp".

默认: NO

listen

如果启用, vsftpd 将以独立模式运行. 这就意味着 vsftpd 不能由类 inetd 来启动. vsftpd 应当直接执行. 由 vsftpd 自身监听和处理联接请求.

默认: NO

listen_ipv6

如 listen 参数, 所不同的是, vsftpd 将对 IPv6 接口进行监听, 而不是 IPv4 接口. 此参数 和 listen 参数相互独立.

默认: NO

local_enable

用于控制是否允许本地登录. 如果启用, /etc/passwd 中的普通帐号即可用于登录.

默认: NO

log_ftp_protocol

如果启用, 假若选项 xferlog_std_format 没有启用, 所有的 FTP 请求和应答都会被记录. 此选项对将对调试很有用.

默认: NO

ls_recurse_enable

如果启用, 此设置将允许用户使用 "ls -R". 这有点安全威胁, 因为在大型站点的根目录下进行 ls -R 将会消耗很多资源.

默认: NO

no_anon_password

如果启用, 匿名用户登录将不再需要密码 - 可以直接登录.

默认: NO

no_log_lock

如果启用, 在写日志文件时, 将会阻止 vsftpd 使用文件锁定. 这个选项通常不会启用. 它的存在是为了处理操作系统的一个 bug, 如 Solaris / Veritas 文件系统组合某些情况下试图锁定日志文件的现象.

默认: NO

one_process_model

如果你使用 Linux 2.4 内核, 您就可以使用一个不同的安全模式, 它只允许每个联接使用一个进程. 这有一点小小的安全问题, 但是提高了性能. 如果您不清楚后果, 或者您的站点要承受大量的并发用户联接时, 请不要启用此选项.

默认: NO

passwd_chroot_enable

如果启用, 同 chroot_local_user 一起使用, 就会基于每个用户创建限制目录, 每个用户限制的目录源于 /etc/passwd 中的家目录. 当家目录路径中包含 /./ 时, enotes that the jail is at that particular location in the path.

pasv_enable

如果数据传输时, 您不允许使用 PASV 模式, 则将此选项设为 NO

默认: YES

pasv_promiscuous

如果您要禁用 PASV 安全检查, 将此选项设置为 YES. 该检查用于确保数据传输联接与控制联接源于同一 IP 地址. 如果不清楚后果, 请不要启用此选项! 此选项只有在某些使用安全隧道的方案中才能正常使用, 或者需要 FXP 的支持.

默认: NO

port_enable

如果您不允许使用端口模式获取数据联接, 将此选项设置为 NO.

默认: YES

port_promiscuous

如果您想禁用端口安全检查, 将此选项设置为 YES. 此检查用于确认出站的数据只流向客户端. 搞清楚后果前,不要启用此选项!

默认: NO

run_as_launching_user

如果您希望可以由用户来启动 vsftpd, 将此选项设置为 YES. 当不能使用 root 登录时, 这通常很有用. 严重警告: 搞清楚后果前,不要启用此选项, 随意的启用此选项将会导致非常严重的安全问题. 特别是 vsftpd 没有/不能 使用目录限制技术来限制文件访问时(甚至 vsftpd 是由 root 启动的). 一个愚蠢的替代方法是将选项 deny_file 设为 {/*,*..*}, 但是其可靠性并不能和限制目录相比, 甚至不在一个等级上. 如果启用此选项, 应当限制其它选项的使用. 例如, 非匿名登录, 上载

文件宿主转换, 使用源自端口 20 的联接和低于 1024 的端口不会工作. 其它一些选项也可能受到影响.

默认值: NO

secure_email_list_enable

如果您要为匿名用户指定一个做为密码的邮件地址列表, 将此选项设置为 YES. 在不需要创建虚拟用户的情况下, 构建一个低安全性访问控制很有用. 如果启用, 匿名用户只有使用在 email_password_file 中指定的邮件地址做为密码, 才能登录. 文件格式是每行一个密码, 没有空格. 默认文件名是 /etc/vsftpd.email_passwords.

默认: NO

session_support

此选项用于控制 vsftpd 是否为登录保持会话. 如果保持会话, vsftpd 将会尝试和更新 utmp 和 wtmp. 如果使用 PAM 认证, 同时还会打开 pam_session, 直至登出. 如果不需要保持登录会话, 或许您希望禁用此选项, 以使得 vsftpd 占用更少的进程和/或更少的特权. 注意 - utmp 和 wtmp 只有在启用 PAM 的情况下才支持.

默认: NO

setproctitle_enable

如果启用, vsftpd 将会尝试在系统进程列表中显示会话状态信息. 也就是说, 进程报告将会显示每个 vsftpd 会话在做什么 (闲置, 下载 等等). 出于安全的考虑, 您可能需要将其关闭.

默认: NO

ssl_enable

如果启用此选项, 并在编译时加入 OpenSSL 支持, vsftpd 将支持通过 SSL 进行安全联接. 此选项用于控制联接(包括登录) 以及数据联接. 您可能同时需要支持 SSL 的客户端. 注意!! 小心启用此选项. 仅在需要时才启用. vsftpd 对使用 OpenSSL 库的安全性不做任何担保. 启用此选项, 就意味着您相信所安装的 OpenSSL 库的安全性.

默认: NO

ssl_sslv2

只有激活 ssl_enable 选项时才有效. 如果启用, 此选项将允许使用 SSL v2 协议进行联接. TLS v1 仍为首选联接.

默认: NO

ssl_sslv3

只有激活 ssl_enable 选项时才有效. 如果启用, 此选项将允许使用 SSL v3 协议进行联接. TLS v1 仍为首选联接.

默认: NO

ssl_tlsv1

只有激活 ssl_enable 选项时才有效. 如果启用, 此选项将允许使用 TLS v1 协议进行联接. TLS v1 仍为首选联接.

默认: YES

syslog_enable

如果启用, 任何本来应该输出到 /var/log/vsftpd/vsftpd.log 的日志, 将会输出到系统日志中. 记录由 FTPD 完成.

默认: NO

tcp_wrappers

如果启用, 并且在编译 vsftpd 时加入了对 TCP_Wrappers 的支持, 则连入请求转由 TCP_Wrappers 完成访问控制. 另外, 这是基于每个 IP 的配置机制. 如果 tcp_wrappers 设置了 VSFTPD_LOAD_CONF 环境变量, 则 vsftpd 会话将会试图加载在此变量中指定的 vsftpd 配置文件.

默认: NO

text_userdb_names

默认情况下, 目录列取时在用户和组字段显示的是数字 ID. 如果启用此选项,则可以得到文本名称. 基于性能的考虑, 默认情况下关闭此选项.

默认: NO

tilde_user_enable

如果启用, vsftpd 将试图解析类似 ~chris/pics 的路径名, 即跟着用户名的波型号. 注意, vsftpd 有时会一直解析 ~ 和 ~/ (这里, ~ 被解析称为初始登录路径). ~user 则只有在可以找到包含闲置目录的 /etc/passwd 文件时才被解析.

默认值: NO

use_localtime

如果启用, vsftpd 在列取目录时, 将显示您本地时区的时间. 默认显示为 GMT. 由 MDTM FTP 命令返回的时间同样也受此选项的影响.

默认: NO

use_sendfile

一个内部设定，用于测试在您的平台上使用 sendfile() 系统的性能.

默认: YES

userlist_deny

此选项只有在激活 userlist_enable 时才会有效. 如果您将此选项设置为 NO, 则只有在 userlist_file 文件中明确指定的用户才能登录系统. 当登录被拒绝时, 拒绝发生在被寻问命令之前.

默认: YES

userlist_enable

如果启用, vsftpd 将会从 userlist_file 选项指定的文件中加载一个用户名列表. 如果用户试图使用列表中指定的名称登录, 那么他们将在寻问密码前被拒绝. 这有助于阻止明文传送密码. 详见 userlist_deny.

默认: NO

virtual_use_local_privs

如果启用, 虚拟用户将拥有同本地用户一样的权限. 默认情况下, 虚拟用户同匿名用户权限相同, 这倾向于更多限制 (特别是在写权限上).

默认: NO

write_enable

用于控制是否允许 FTP 命令更改文件系统. 这些命令是: STOR, DELE, RNFR, RNTO, MKD, RMD, APPE 和 SITE.

默认: NO

xferlog_enable

如果启用, 将会维护一个日志文件, 用于详细记录上载和下载. 默认情况下, 这个日志文件是 /var/log/vsftpd.log. 但是也可以通过配置文件中的 vsftpd_log_file 选项来指定.

默认: NO(但是在示例设置中启用了这个选项)

xferlog_std_format

如果启用, 传输日志文件将以标准 xferlog 的格式书写, 如同 wu-ftpd 一样. 这可以用于重新使用传输统计生成器. 然而, 默认格式更注重可读性. 此格式的日志文件默认为 /var/log/xferlog, 但是您也可以通过 xferlog_file 选项来设定.

默认: NO

数字选项

下边是数字选项的列表. 数字选项必须设置一个非负的整数. 为了便于 umask 选项, 同样也支持八进制数字. 八进制数字首位应为 0 .

accept_timeout

超时, 以秒计, 用于远程客户端以 PASV 模式建立数据联接.

默认: 60

anon_max_rate

允许的最大数据传输速率, 单位 b/s, 用于匿名客户端.

默认: 0 (无限制)

anon_umask

用于设定匿名用户建立文件时的 umask 值. 注意! 如果您要指定一个八进制的数字, 首位应当是 "0", 否则将视作 10 进制数字.

默认: 077

connect_timeout

超时, 单位 秒, 用于响应 PORT 方式的数据联接.

默认: 60

data_connection_timeout

超时, 单位 秒, 用于设定空闲的数据连接所允许的最大时长. 如果触发超时, 则远程客户端将被断开.

默认: 300

file_open_mode

用于设定创建上载文件的权限. mask 的优先级高于这个设定. 如果想允许上载的文件可以执行, 将此值修改为 0777

默认: 0666

ftp_data_port

FTP PORT 方式的数据联接端口.(需要激活 connect_from_port_20 选项)

默认: 20

idle_session_timeout

超时, 单位 秒, 远程客户端的最大 FTP 命令间隔. 如果超时被触发, 远程客户端将被断开.

默认: 300

listen_port

如果 vsftpd 以独立模式启动, 此端口将会监听 FTP 连入请求.

默认: 21

local_max_rate

允许的最大数据传输速率, 单位 b/s, 用于限制本地授权用户.

默认: 0 (无限制)

local_umask

用于设定本地用户上载文件的 umask 值. 注意! 如果您要指定一个八进制的数字, 首位应当是 "0", 否则将视作 10 进制数字.

默认: 077

max_clients

如果 vsftpd 以独立模式启动, 此选项用于设定最大客户端联接数. 超过部分将获得错误信息.

默认: 0 (无限制)

max_per_ip

如果 vsftpd 以独立模式启动, 此选项用于设定源于同一网络地址的最大联接数. 超过部分将获得错误信息.

默认: 0 (无限制)

pasv_max_port

为 PASV 方式数据联接指派的最大端口. 基于安全性考虑, 可以把端口范围指定在一样较小的范围内.

默认: 0 (可以使用任意端口)

pasv_min_port

为 PASV 方式数据联接指派的最小端口. 基于安全性考虑, 可以把端口范围指定在一样较小的范围内.

默认: 0 (可以使用任意端口)

trans_chunk_size

您可能不想修改这个设置, 如果有带宽限制, 可以尝试将此值设置为 8192.

默认: 0 (让 vsftpd 自己选择一个更合理的设置)

字符选项

下边是字符选项列表

anon_root

此选项声明, 匿名用户在登录后将被转向一个指定目录(译者注: 默认根目录). 失败时将被忽略.

默认: (无)

banned_email_file

此选项用于指定包含不允许用作匿名用户登录密码的电子邮件地址列表的文件. 使用此选项需要启用 deny_email_enable 选项.

默认: /etc/vsftpd.banned_emails

banner_file

此选项用于指定包含用户登录时显示欢迎标识的文件. 设置此选项, 将取代 ftpd_banner 选项指定的欢迎标识.

默认: (无)

chown_username

用于指定匿名用户上载文件的宿主. 此选项只有在 chown_uploads 选项设定后才会有效.

默认；root

chroot_list_file

此选项用于指定包含被限制在家目录中用户列表的文件. 使用此选项, 需启用 chroot_list_enable . 如果启用了 chroot_local_user 选项, 此文件所包含的则为不会被限制在家目录中的用户列表.

默认: /etc/vsftpd.chroot_list

cmds_allowed

此选项指定允许使用的 FTP 命令(登录以后. 以及登录前的 USER, PASS 和 QUIT), 以 逗号分割. 其它命令将被拒绝使用. 这对于锁定一个 FTP 服务器非常有效. 例如: mds_allowed=PASV,RETR,QUIT

默认: (无)

deny_file

此选项用于设定拒绝访问的文件类型(和目录名等). 此设定并不是对文件进行隐藏, 但是您不能对其操作(下载, 更换目录, 以及其它操作). 此选项非常简单, 不能用于严格的访问控制--文件系统的优先级要高一些. 然而, 此选项对于某些虚拟用户的设定非常有效. 特别是在一个文件可以通过各种名称访问时(可能时通过符号联接或者硬联接), 应当注意拒绝所有的访问方法. 与 hide_file 中给出名称匹配的文件会被拒绝访问. 注意 vsftpd 只支持正则表达式匹配的部分功能. 正因为如此, 您需要尽可能的对此选项的设置进行测试. 同时基于安全性考虑, 建议您使用文件系统自身的访问控制. 例如: deny_file={*.mp3,*.mov,.private}

默认: (无)

dsa_cert_file

此选项用于指定用于 SSL 加密联接的 DSA 证书的位置.

默认: (无 - 使用 RSA 证书)

email_password_file

此选项用于提供启用 secure_email_list_enable 选项, 所需要的可替代文件.

默认: /etc/vsftpd.email_passwords

ftp_username

用于处理匿名 FTP 的用户名. 此用的家目录即为匿名发 FTP 的根目录.

默认: ftp

ftpd_banner

用于替换首次连入 vsftpd 时显示的欢迎标识字符串.

默认: (无 - 显示默认 vsftpd 标识)

guest_username

参阅布尔选项 guest_enable . 此选项用于将 guest 用户映射到一个真实用户上.

默认: ftp

hide_file

此选项用于设定列取目录时, 要隐藏的文件类型(以及目录等). 尽管隐藏了, 知道其宿主的客户端仍然能对文件/目录等有完全访问权限. 与名称 hide_file 中包含的字符串匹配的项都将隐藏. 注意 vsftpd 只支持正则表达式匹配的部分功能. 例如: hide_file={*.mp3,.hidden,hide*,h?}

默认: (无)

listen_address

如果 vsftpd 以独立模式运行, 此设定用于修改默认(所有本地接口)监听地址. 格式为数字 IP 地址.

默认: (无)

listen_address6

如 listen_address, 不过应该指定为 IPv6 监听器指定默认监听地址. 格式为标准 IPv6 地址格式.

默认: (无)

local_root

本选项用于指定本地用户(即, 非匿名用户)登录后将会转向的目录. 失败时将被忽略.

默认: (无)

message_file

此选项用于指定进入新目录时要查询的文件名. 这个文件的内容为显示给远程用户的欢迎信息. 使用此选项, 需要启用 dirmessage_enable 选项.

默认: .message

nopriv_user

用于指定一个用户, 当 vsftpd 要切换到无权限状态时, 使用此用户. 注意这最好是一个专用用户, 而不是用户 nobody. 在大多数机器上, 用户 nobody 被用于大量重要的事情.

默认: nobody

pam_service_name

用于指定 PAM 服务的名称.

默认: ftp

pasv_address

此选项为 vsftpd 指定一个 IP 地址, 用作对 PASV 命令的响应. IP 地址应该为数字模式.

默认: (无 - 即地址从连入的联接套接字中获取)

rsa_cert_file

此选项用于指定 SSL 加密联接所用 RSA 证书的位置.

默认: /usr/share/ssl/certs/vsftpd.pem

secure_chroot_dir

此选项用于指定一个空目录. 并且 ftp 用户不应对此目录有写权限. 当 vsftpd 不需要访问文件系统是, 此此目录做为一个限制目录, 将用户限制在此目录中.

默认: /usr/share/empty

ssl_ciphers

此选项用于选择 vsftpd 允许使用哪些 SSL 加密算法来用于 SSL 加密联接. 更多信息参阅 ciphers 的联机手册. 注意这样可以有效的防止对某些发现漏洞的算法进行恶意的远程攻击.

默认: DES-CBC3-SHA

user_config_dir

此选项用于定义用户个人配置文件所在的目录. 使用非常简单, 一个例子即可说明. 如果您将 user_config_dir 设置为 /etc/vsftpd_user_conf 并以用户 "chris"登录, 那么 vsftpd 将对此用户使用文件 /etc/vsftpd_user_conf/chris 中的设定. 此文件的格式在联机手册中有详细说明. 请注意, 不是每个设定都能影响用户的. 例如, 有些设定只在用户会话开始时起作用. 这包括 listen_address, banner_file, max_per_ip, max_clients, xferlog_file, 等等.

默认: (无)

user_sub_token

此选项需要和虚拟用户联合使用. 其将依据一个模板为每个虚拟用户创建家目录. 例如, 如果真实用户的家目录由选项 guest_username 指定为 /home/virtual/$USER, 并且将 user_sub_token 设定为 $USER, 当虚拟用户 fred 登入后, 将会进入(限制)目录 /home/virtual/fred. 如果 local_root 中包含了 user_sub_token 此选项也会起作用.

默认: (无)

userlist_file

此选项用于指定启用 userlist_enable 选项后需要加载文件的名称.

默认: /etc/vsftpd.user_list

vsftpd_log_file

此选项用于指定写入 vsftpd 格式日志的文件. 如果启用了 xferlog_enable, 而没有设定 xferlog_std_format 的话, 日志将只会写入此文件. 另为,如果设置了 dual_log_enable 的话, 日志同样会写入此文件. 更复杂一点, 如果您设置了 syslog_enable 的话, 输出将不会写入此文件, 而是写入系统日志文件.

默认: /var/log/vsftpd.log

xferlog_file

此选项用于指定写入 wu-ftpd 样式日志的文件名. 只有在 xferlog_enable 和 xferlog_std_format 做了相应设定, 才会记录此传输日志. 另外, 如果您设置了 dual_log_enable 选项, 也会记录此日志.

默认: /var/log/xferlog

739 ☆更改 FTP PORT

将预设的 port 21 更换为 2121
修改/etc/vsftpd/vsftpd.conf
新增底下一行
listen_port=2121
重新启动 vsftpd

740 ☆容许 root 用户远程登陆

修改/etc/vsftpd.ftpusers 和/etc/vsftpd.user_list(如果 userlist_deny=YES.,这个是缺省值)
将 root 一栏删除或在前头加"#"注释;

741 ☆匿名权限

若是读者的主机不希望使用者匿名登入，则可参考以下步骤。
修改/etc/vsftpd/vsftpd.conf
将
anonymous_enable=YES
改为
anonymous_enable=NO
重新启动 vsftpd

首先:我们要改一下 vsftpd.conf，确保有以下几行;
anonymous_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_umask=022 (默认是 077,上传后就无法看了)

其次:在 ftp 用户家目录的下建一个文件夹，并修改其权限为完全开放;

ftp 用户的家目录在哪？我们前面已经说了，要通过/etc/passwd 来查看;也可以通过 finger ftp 来查看;
[root@panda ~]# finger ftp
Login: ftp Name: FTP User
Directory: /var/ftp Shell: /sbin/nologin

这说明 ftp 用户的家目录在/var/ftp ，我们要在这个目录下建一个目录，然后把他的权限设置为任何用户可读可写可执行就行了;一般的情况下，在发行版中，有一个/var/ftp/pub 的目录，如果没有，您也可以自己建一个;把配置文件改好后，只要把/var/ftp 下的任何一个目录的权限打开，都可以用来匿名上传和下载;

比如您想让匿名用户上传和下载都在/var/ftp/pub，就可以把/var/ftp/pub 的权限打开，如果没有这个目录，您要自己建一个;
[root@localhost ~]# mkdir /var/ftp/pub
[root@localhost ~]# chmod 777 /var/ftp/pub
这样上传的时候传到 pub 目录就 OK 了，对不对？？

742 ☆本地权限

FTP 用户一般是不能登录系统的，这也是为了安全。在系统中，没有权限登录系统的用户一般也被称之为虚拟用户;虚拟用户也是要写进/etc/passwd 中;这只是一种虚拟用户的方法，但说实在的并不是真正的虚拟用户，只是把他登录 SHELL 的权限去掉了，所以他没有能力登录系统;

本地用户用 ftp 会登录到自己的 home 目录中

如果我们想把 panda 这个用户目录定位在/var/panda 这个目录中，并且不能登录系统;我们应该如下操作
[root@localhost ~]# adduser -d /var/panda -g ftp -s /sbin/nologin panda
[root@localhost ~]# passwd panda
Changing password for user panda.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.

其实这还是不够的，还要改一下配置文件 vsftpd.conf，以确保本地虚拟用户能有读写权限;
local_enable=YES
write_enable=YES
local_umask=022 (默认是 077)

743 ☆更改 FTP 目录

对于 ftp 这个用户的管理，我们应该查看/etc/passwd ，然后修改 ftp 用户那行;
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

比如我们想把 ftp 用户的家目录改为/opt/ftp，则要把类似上一行改为
ftp:x:14:50:FTP User:/opt/ftp:/sbin/nologin

然后我们要建立 ftp 用户的新的家目录;
[root@localhost ~]# mkdir /opt/ftp
[root@localhost ~]# chmod 755 /opt/ftp
[root@localhost ~]# chown root:root /opt/ftp

如果默认的 ftp 磁盘空间紧张，我们其实也可以用虚拟路径映射的方法来解决;也就是 mount --bind 的办法;看情况吧，哪个适合就是最好的方法;

744 ☆实现虚拟路径

比如:
/home/a 映射为 ftp://localhost/a
/home/b/c 则为 ftp://localhost/c

我们可以通过如下的方法来实现。
[root@localhost ~]# mount --bind [原有的目录] [新目录]

比如我的 ftp 的默认目录是/var/ftp，我想把/mnt/rpms 文件夹，映射到/var/ftp/rpms 目录中，我就如下操作

然后执行 mount 命令
[root@localhost ~]# mount --bind /mnt/rpms /var/ftp/rpms
这样就 OK 了。

745 ☆定制欢迎信息

实现这个并不难，我们可以查看 vsftpd.conf 文件中，是否有这行。
dirmessage_enable=YES
message_file=.message

如果没有就加上，如果 dirmessage_enable=YES 前面有#号，就把#号去掉。
其实 FTP 信息默认的就是.message，所以可以不加 message_file= 来指定。自己指定也行，无所谓的事;

然后我们制定一个.message 文件，写上您想要写的东西
我们可以用编辑器来写这个.message，我想这个过程就不用说了吧。
hello to panda's vsftp

接着，在各目录之中，新增名为.message 的文件
步骤同上
第一次切换到这个目录的时候会有信息显示
250-hello to panda's vsftp

746 ☆配置 vsFTP 限制

修改/etc/vsftpd/vsftpd.conf

限制链接
max_clients=数字
max_per_ip=数字

让我的 vsFTP 最大支持链接数为 100 个，每个 IP，最多能支持 5 个链接，所以我应该在 vsftpd.conf 中加上如下的两行:
max_clients=100
max_per_ip=5

限制传输速度
anon_max_rate=数字
注:这是匿名的速度
local_max_rate=数字
注:这是 vsFTP 服务器上本地用户的速度

所以我们要在 vsftpd.conf 中加入下面的两行
anon_max_rate=81920
local_max_rate=81920
让匿名用户和 vsFTP 上的用户都以 80KB 下载

747 ☆针对不同的使用者的限制

修改/etc/vsftpd/vsftpd.conf
新增底下一行
user_config_dir=/etc/vsftpd_user_conf

新增一个目录:/etc/vsftpd_user_conf
mkdir /etc/vsftpd_user_conf

在/etc/vsftpd_user_conf 之下新增一个名为用户名的文件
编辑限制内容;

如/etc/vsftpd_user_conf/panda 文件就是限制 panda 用户的

748 ☆把用户限制在 home 中

把系统内所有的 FTP 用户都限制在家目录中
我们可以通过更改 vsftpd.conf 文件，加入如下的一行
chroot_local_user=YES
改完配制文件，不要忘记重启 vsFTPd 服务器;

也可以选择性的限制用户都限制在家目录中
我们要自己建一个文件，在/etc 目录中或/etc/vsftpd 目录中
#touch /etc/vsftpd.chroot_list
以 panda 和 wawa 这两个用户限制在他们所在的家目录中，而其它的 FTP 用户不做此限制。
在 vsftpd.chroot_list 这个文件中，把 panda 和 wawa 添上去就行，注意，每个用户占一行。
panda
wawa

然后改/etc/vsftpd/vsftpd.conf 文件，找如下的两行
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list #原文是 chroot_list

如果没有这样的两行，就可以自己添加上去也是一样的。
设置好后，重新 vsFTPD 服务器。
ftp> cd /home
550 Failed to change directory. #切换错误

并且会显示先对路径,不是绝对路径了

749 ☆绑定 IP 到 vsFTPd

如何让绑定 IP 到 vsFTPd？也就是说，如何让用户只能通过某个 IP 来访问 FTP。其实这个功能很有意思。如果绑定的是内网的 IP，外部是没有办法访问的。如果绑定的是对外服务的 IP，内网也只能通过对外服务的 IP 来访问 FTP
listen_address=192.168.0.2

750 ☆vsFTPd 的日志

把下面 xferlog_file 前面的#号对掉，也就是把 vsftp 的 log 功能打开，这样我们就能在/var/log 目录下查看 vsftpd.log。这是 vsFTP 的日志功能，这对于我们来说是极为重要的。
xferlog_file=/var/log/vsftpd.log

751 ☆显示 vsftp 的每一个 process

一般启动 vsftp 时，我们只会看到一个名为 vsftpd 的 process 在运作，但若是读者希望每一个联机，都能以独立的 process 来呈现，则可执行以下步骤。
修改/etc/vsftpd/vsftpd.conf

新增底下一行
setproctitle_enable=YES
重新启动 vsftpd

使用 ps -ef 的指令
[root@home vsftpd]# ps -ef|grep ftp
root 2090 1 0 16:41 pts/0 00:00:00 vsftpd: LISTENER
nobody 2120 2090 0 17:18 ? 00:00:00 vsftpd: 192.168.10.244:connected
test1 2122 2120 0 17:18 ? 00:00:00 vsftpd: 192.168.10.244/test1:IDLE
nobody 2124 2090 0 17:19 ? 00:00:00 vsftpd: 192.168.10.244:connected
test2 2126 2124 0 17:19 ? 00:00:00 vsftpd: 192.168.10.244/test2:IDLE
root 2129 1343 0 17:20 pts/0 00:00:00 grep ftp

752 ☆vsftpd 与 TCP_wrapper 结合

确定/etc/vsftpd/vsftpd.conf 之中 tcp_wrappers 的设定为 YES
tcp_wrappers=YES

753 ☆vsFTPd 添加虚拟用户

通过 pam 认证，用 db_load 添加用户，是真正的虚拟用户。现在我们简单的介绍一下，通过以后的学习，我们再深入补充:

在/etc/pam.d/目录中创建一个文件 ftp

在/etc/pam.d/ftp 里面加上如下的两行
auth required /lib/security/pam_userdb.so db=/etc/vsftpd_login
account required /lib/security/pam_userdb.so db=/etc/vsftpd_login

创建一系统的用户名用密码的文件 logins.txt
里面是 FTP 的虚拟用户名和密码:
panda07
123456
panda08
234567
panda09
567890
panda10
678901
panda11
789012

创建一个真实的用户名 panda06

把/etc/hosts 复制到/home/panda06，并改变它的属主
[root@panda01 root]#cp /etc/hosts /home/panda06/hosts
[root@panda01 root]#chown panda06.panda06 /home/panda06/hosts

通过 db_load 来创建虚拟用户的库文件。
[root@panda01 root]# db_load -T -t hash -f logins.txt /etc/vsftpd_login.db

更改 vsftpd.conf 文件，加入如下的几行
pam_service_name=ftp
guest_enable=YES
guest_username=panda06
anon_world_readable_only=NO

重启 vsFTPd 服务器;

754 ☆FTP 错误

如果我们已经把 vsFTPd 服务器启动好了，但登录测试是会出现类似下面的提示;
500 OOPS: vsftpd: refusing to run with writable anonymous root
这表示 ftp 用户的家目录的权限不对，应该改过才对;
[root@localhost ~]# more /etc/passwd |grep ftp
ftp:x:1000:1000:FTP User:/var/ftp:/sbin/nologin
我们发现 ftp 用户的家目录在/var/ftp，就是这个/var/ftp 的权限不对所致，这个目录的权限是不能打开所有权限的;是您运行了 chmod 777 /var/ftp 所致;如果没有 ftp 用户这个家目录，当然您要自己建一个;

如下 FTP 用户的家目录是不能针对所有用户、用户组、其它用户组完全开放;
[root@localhost ~]# ls -ld /var/ftp
drwxrwxrwx 3 root root 4096 2005-03-23 /var/ftp

修正这个错误，应该用下面的办法;
[root@localhost ~]# chown root:root /var/ftp
[root@localhost ~]# chmod 755 /var/ftp

有的弟兄可能会说，那匿名用户的可读、可下载、可上传怎么办呢？这也简单，在/var/ftp 下再建一个目录，权限是 777 的就行了，再改一改 vsftpd.conf 就 OK 了;没有什么难的;

vsFTPd 出于安全考虑，是不准让 ftp 用户的家目录的权限是完全没有限制的，您可以去读一下 vsFTPd 的文档就明白的了;否则也不能称为最安全的 FTP 服务器了，对不对？

755 ☆FTP 命令

FTP 命令是 Internet 用户使用最频繁的命令之一,不论是在 DOS 还是 UNIX 操作系统下 使用 FTP,都会遇到大量的 FTP 内部命令,熟悉并灵活应用 FTP 的内部命令,可以大大方便 使用者,对于现在拨号上网的用户，如果 ISP 提供了 shell 可以使用 nohup，那么 ftp 将是 你最省钱的上 download 方式

ftp 的命令行格式为:ftp -v -d -i -n -g[主机名]

-v 显示远程服务器的所有响应信息。

-d 使用调试方式。

-n 限制 ftp 的自动登录,即不使用.netrc 文件。

-g 取消全局文件名。

ftp 使用的内部命令如下(其中括号表示可选项):

1.![cmd[args]]在本地机中执行交互 shell、exit 回到 ftp 环境,如!ls＊.zip。

2.￥ macro-ame[args]执行宏定义 macro-name。

3.account[password]提供登录远程系统成功后访问系统资源所需的补充口令。

4.appendlocal-file[remote-file]将本地文件追加到远程系统主机,若未指定远程系统文件名,则使用本地文件名。

5.ascii 使用 ascii 类型传输方式。

6.bell 每个命令执行完毕后计算机响铃一次。

7.bin 使用二进制文件传输方式。

8.bye 退出 ftp 会话过程。

9.case 在使用 mget 时,将远程主机文件名中的大写转为小写字母。

10.cd remote-dir 进入远程主机目录。

11.cdup 进入远程主机目录的父目录。

12.chmod modefile-name 将远程主机文件 file-name 的存取方式设置为 mode,如 chmod 777 a.out。

13.close 中断与远程服务器的 ftp 会话(与 open 对应)。

14.cr 使用 asscii 方式传输文件时,将回车换行转换为回行。

15.delete remote-file 删除远程主机文件。

16.debug[debug-value]设置调试方式,显示发送至远程主机的每条命令,如 debug3,若 设为 0,表示取消 debug。

17.dir[remote-dir][local-file]显示远程主机目录,并将结果存入 local-file。

18.disconnection 同 close。

19.form format 将文件传输方式设置为 format,缺省为 file 方式。

20.getremote-file[local-file]将远程主机的文件 remote-file 传至本地硬盘的 local-file。

21.glob 设置 mdelete、mget、mput 的文件名扩展,缺省时不扩展文件名,同命令行的-g 参数。

22.hash 每传输 1024 字节,显示一个 hash 符号(＃)。

23.help[cmd]显示 ftp 内部命令 cmd 的帮助信息,如 help get。

24.idle[seconds]将远程服务器的休眠计时器设为[seconds]秒。

25.image 设置二进制传输方式(同 binary)

26.lcd[dir]将本地工作目录切换至 dir。

27.ls[remote-dir][local-file]显示远程目录 remote-dir,并存入本地 local-file。

28.macdef macro-name 定义一个宏,遇到 macdef 下的空行时,宏定义结束。
29.mdelete[remote-file]删除远程主机文件。
30.mdir remote-files local-file 与 dir 类似,但可指定多个远程文件,如 mdir＊.o.＊. zipoutfile。
31.mget remote-files 传输多个远程文件。
32.mkdir dir-name 在远程主机中建一目录。
33.mls remote-file local-file 同 nlist,但可指定多个文件名。
34.mode[mode-name]将文件传输方式设置为 mode-name,缺省为 stream 方式。
35.modtime file-name 显示远程主机文件的最后修改时间。
36.mput local-file 将多个文件传输至远程主机。
37.newerfile-name 如果远程机中 file-name 的修改时间比本地硬盘同名文件的时间更近,则重传该文件。
38.nlist[remote-dir][local-file]显示远程主机目录的文件清单,并存入本地硬盘的 local-file。
39.nmap[inpatternoutpattern]设置文件名映射机制,使得文件传输时,文件中的某些字符相互转换,如 nmap￥1.￥2.￥3[￥1,￥2].[￥2,￥3],则传输文件 a1.a2.a3 时 ,文件名变为 a1、a2，该命令特别适用于远程主机为非 U-NIX 机的情况。
40.ntrans[inchars[outchars]]设置文件名字符的翻译机制,如 ntrans1R,则文件名 LL L 将变为 RRR。
41.open host[port]建立指定 ftp 服务器连接,可指定连接端口。
42.passive 进入被动传输方式。
43.prompt 设置多个文件传输时的交互提示。
44.proxyftp-cmd 在次要控制连接中,执行一条 ftp 命令,该命令允许连接两个 ft p 服务器,以在两个服务器间传输文件。第一条 ftp 命令必须为 open,以首先建立两个服务器间的连接。
45.put local-file[remote-file]将本地文件 local-file 传送至远程主机。
46.pwd 显示远程主机的当前工作目录。
47.quit 同 bye,退出 ftp 会话。
48.quote arg1,arg2……将参数逐字发至远程 ftp 服务器,如 quote syst。
49.recv remote-file[local-file]同 get。
50.regetremote-file[local-file]类似于 get,但若 local-file 存在,则从上次传输中断处续传。
51.rhelp[cmd-name]请求获得远程主机的帮助。
52.rstatus[file-name]若未指定文件名,则显示远程主机的状态,否则显示文件状态。
53.rename[from][to]更改远程主机文件名。
54.reset 清除回答队列。
55.restart marker 从指定的标志 marker 处,重新开始 get 或 put,如 restart 130。
56.rmdir dir-name 删除远程主机目录。
57.runique 设置文件名唯一性存储,若文件存在,则在原文件后加后缀。
58.send local-file[remote-file]同 put。
59.sendport 设置 PORT 命令的使用。
60.site arg1,arg2……将参数作为 SITE 命令逐字发送至远程 ftp 主机。
61.size file-name 显示远程主机文件大小,如 site idle 7200。
62.status 显示当前 ftp 状态。
63.struct[struct-name]将文件传输结构设置为 struct-name,缺省时使用 stream 结构。
64.sunique 将远程主机文件名存储设置为唯一(与 runique 对应)。
65.system 显示远程主机的操作系统类型。

66.tenex 将文件传输类型设置为 TENEX 机所需的类型。
67.tick 设置传输时的字节计数器。
68.trace 设置包跟踪。
69.type[type-name]设置文件传输类型为 type-name,缺省为 ascii,如 typebinary,设置二进制传输方式。
70.umask[newmask]将远程服务器的缺省 umask 设置为 newmask,如 umask 3。
71.useruser-name[password][account]向远程主机表明自己的身份,需要口令时,必须输入口令,如 user anonymous my@email。
72.verbose 同命令行的-v 参数,即设置详尽报告方式,ftp 服务器的所有响应都将显示给用户,缺省为 on.
73.?[cmd]同 help。

那么如何应用这些命令提高效率呢？下面我举一个例子，如何利用 fttp 进行后台下载，假设你的 ISP 给你提供了 shell 并且可以用 nohup，你想由 fttp.download.com/pub/internet/ 下载一个 30M 的程序 aaa.zip 具体步骤如下:
1.用 notepad 做一个文件如 aaa1 内容如下
open ftp.dwonload.com
user anonymous zyz@cenpok.net
cd /pub/internet/
i
get aaa.zip
close
bye
2.拨号登录到你的 ISP 上。用 telnet 或 netterm 登录到 shell，一般都在你的 home 子目录里 bbs~/
3.用 fttp 上传 aaa1 到 ISP 服务器你的子目录。
4. 执行 nohup fttp -invd aaa2＆
这样这个进程就被放在 ISP 服务器的后台进行了，如果你想知道情况如何，可以 more aaa2 就可以知道情况如何了。这时你可以断线了或干点别的，估计时间到了(time 约=30M/(33.6K/9)s)拨号上去，more aaa2 如果显示成功下载 aaa.zip 就表示 aaa.zip 已经被下载到 ISP 的服务器上了，你再由 ISP 的服务器拉回来就相当与点对点了,记得下载完成后 del 掉你的文件(aaa.zip)，免得浪费 ISP 资源，它会关掉 shell 的。

756 ☆FTP 数字代码的意义

110 重新启动标记应答。
120 服务在多久时间内 ready。
125 数据链路端口开启，准备传送。
150 文件状态正常，开启数据连接端口。
200 命令执行成功。
202 命令执行失败。
211 系统状态或是系统求助响应。
212 目录的状态。

213 文件的状态。
214 求助的讯息。
215 名称系统类型。
220 新的联机服务 ready。
221 服务的控制连接端口关闭，可以注销。
225 数据连结开启，但无传输动作。
226 关闭数据连接端口，请求的文件操作成功。
227 进入 passive mode。
230 使用者登入。
250 请求的文件操作完成。
257 显示目前的路径名称。
331 用户名称正确，需要密码。
332 登入时需要账号信息。
350 请求的操作需要进一部的命令。
421 无法提供服务，关闭控制连结。
425 无法开启数据链路。
426 关闭联机，终止传输。
450 请求的操作未执行。
451 命令终止:有本地的错误。
452 未执行命令:磁盘空间不足。
500 格式错误，无法识别命令。
501 参数语法错误。
502 命令执行失败。
503 命令顺序错误。
504 命令所接的参数不正确。
530 未登入。
532 储存文件需要账户登入。
550 未执行请求的操作。
551 请求的命令终止，类型未知。
552 请求的文件终止，储存位溢出。
553 未执行请求的的命令，名称不正确。

757 ☆bc

简单的计算器

758 ☆xinetd.conf

/etc/xinetd.conf
超级守护进程配置文件

defaults

```
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d
```

759☆tcp-wrapper

Tcp wrapper 是 Wietse Venema 的大作。
Linux 提供另外一种更为灵活和有效的方法来实现对服务请求用户的限制，从而可以在保证安全性的基础上，使可信任用户使用各种服务。Linux 提供了一个叫 TCP wrapper 的程序。在大多数发布版本中该程序往往是缺省地被安装。利用 TCP wrapper 你可以限制访问前面提到的某些服务。而且 TCP wrapper 的记录文件记录了所有的企图访问你的系统的行为。通过 last 命令查看该程序的 log，管理员可以获知谁曾经或者企图连接你的系统。

是 xinetd 的防火墙,只对 xinetd 的子服务作用(除非其他服务支持 tcp-wrapper,并打开 tcp-wrapper 认证)

不是任何服务程序都能使用 TCP_wrappers 的，例如使用命令 ldd /usr/sbin/sshd，如果输出中有 libwrap，则说明可以使用 TCP_wrappers, 即该服务可以使用/etc/hosts.allow 和/etc/hosts.deny,如果输出没有 libwrap 则不可使用

在/etc 目录下，有两个文件：hosts.deny hosts.allow 通过配置这两个文件，你可以指定哪些机器可以使用这些服务，哪些不可以使用这些服务。

tcp-wrapper 的判断顺序是先看是否明确允许(hosts.allow)，再看是否明确禁止(hosts.deny)，如果都没有明确，那就是默认允许。
俩个文件均没有此 IP,则访问允许;
俩个文件均有此 IP,则 allow 优先;

/etc/hosts.allow <- 设定允许的清单
/etc/hosts.deny <- 设定拒绝的清单

对于能过 xinetd 程序启动的网络服务，比如 tftp telnet，我们就可以修改/etc/hosts.allow 和 /etc/hosts.deny 的配制，来许可或者拒绝哪些 IP、主机、用户可以访问。

设定的格式:

daemon_list : client_list [:options]

daemon_list 服务名称:
这得在/etc/services 有定义的服务名称
写成在/etc/xinetd.d 目录中的文件中的 service 变量

Client_List 可以使用的表示方式为 :
- by IP Address (192.168.152.128, 10.0.0.)
- by name (www.redhat.com, .panda.com)
- by netmask (192.168.0.0/255.255.255.0 ,192.168. ,但不能是 192.168.0.0/24 这样)
- by network name (from NIS or /etcnetworks ex: @mynetwork)

wildcard 表示方式
ALL : 所有
LOCAL : /etc/hosts 有的主机
UNKNOWN : 未知的主机(没有来自 hosts 或 dns)
KNOWN : 已知的主机(来自 hosts 或 dns)
PARANOID : 正反解内容皆不符合者
EXCEPT : 除此规则外

option 栏位的应用方式 : service_list : client_list [:option1 :option2 ..]
%c : 用户资讯
%s : 主机资讯
%h : 用户 host name or ip
%p : server PID
注意:这两个 options 只能摆在所有的 options list 的最後一个.

基本语法
only_from = host_pattern (only_from = 192.168.0.0/24)
no_access = host_pattern (no_access = 192.168.152.128)
instances = 60 (最多可启动 60 个服务)
per_source = 5 (同一个 IP 只能建立 5 个 Session)
access_times = 08:00-16:00 (开放服务的时间)

备注: 修改完成设定档後要重新启动服务 service xinetd restart

如果一台主机有两个网卡时的设定方式. (可针对网卡使用不同的机制)
in.telnetd@192.168.0.254: 192.168.0.
in.telnetd@192.169.152.128: 192.168.0.

in.telnetd: ALL :/bin/mail -s "temp test" panda@panda.panda.com

默认启动 xinetd 的一个服务都发送到/var/log/message 中,这里可用 spawn 来运行另外一个 shell 命令

in.telnetd : ALL : spawn echo "login attempt from %c to %s " } mail -s warning root
(当有使用者用 telnet 连结到主机时寄发信件给 root)

#/etc/hosts.allow
vsftpd: 192.168.0.
in.telnetd, portmap: 192.168.0.8

#/etc/hosts.deny
ALL .cracker.org EXCEPT trusted.cracker.org
vsftpd, portmap: ALL
pop3d: 192.168.0. EXCEPT 192.168.0.4

ALL: .bad.domain: DENY <--设定这个的 domain 来的机都一概不理.
fingerd: UNKNOWN@ALL: DENY <--设定没有装 ident 的机器都不让它 finger 我们.
ALL: ALL@ALL: ALLOW <所以不是上述情况的机器,都可以接受.

fingerd: UNKNOWN@ALL: banners /usr/daemon/mesg: DENY
telnetd: ALL@ALL: banners /usr/daemon/mesg: ALLOW
ftpd: ALL: rfc931: keepalive: allow
ALL: 140.119.: keepalive: ALLOW
ALL EXCEPT fingerd: ALL: DENY
上面的设定:finger 只能给有装 identd 之类软体的机器使用,拒绝时会有所表示;telnetd 允许全部人使用,且会先秀段讯息,其他的不给 140.119.之外的人使用.

设置好后，要重新启动(没必要)
/etc/rc.d/init.d/xinetd restart
/etc/rc.d/init.d/network restart

在 hosts.allow 中配置为
vsftpd: 192.168.0.0/24 EXCEPT 192.168.0.8

hosts.deny 中配置为
vsftpd: ALL

让 192.168.0.0/24 的网段可以访问 vsftpd ， 排除其他网段，包括 192.168.0.8 的主机

如果 hosts.deny 中不设置的， 192.168.0.8 还是可以 ftp 的,192.168.0.8 在 hosts.allow 中并没有规定明确允许，注意 hosts.allow 只是规定是否允许，不等于 except 了就代表 deny。在 hosts.deny 中没有规定 192.168.0.8 禁止(因为没有写 vsftpd: ALL)，所以 192.168.0.8 最终被允许。

格式
process_name@host_pattern : client_list ...
daemon_list : ... user_pattern@host_pattern ..
都可以
这样
ALL: LOCAL @some_netgroup
ALL: .foobar.edu EXCEPT terminalserver.foobar.edu

要 vsftp 支持 tcp-wrapper
在/etc/vsftpd/vsftpd.conf 中必须有以下 2 行：
listen=YES
tcp_wrappers=YES

760 ☆tcpdchk

TCP Wrappers 设定检查程式
Tcpchck 是确认您 TCP Wappers 的设定是否正确的工具,并报告出所发现之可能的和实际发生的问题,该程式会检查 tcpd 存取控制档(预设为/etc/hosts.allow 与/etc/hosts.deny),并将这些文件中的项目与网路设定档 inetd 或 tlid 中的项目进行比对.

Tcpdchk 分析您的设定的错误
1 语法错误
2 路径名称错误
3 主机或 IP 位址错误
4 主机名称或 IP 位址不符
5 您已指定规则的服务并未包装於 tcpd 之内 Tcpdchk 的参数

-a 指定 tcpdchk 应对不属於 ALLOW 万用字元涵括范围之允许规则提出报告
-i 用来指定 inetd.conf 的替代档
-v 详细的格式化输出

761 ☆tcpdmatch

显示使用规则的结果
tcpdmatch [常驻程式] [主机]
tcpdmatch in.telnetd techsupport.theircompapy.net
TCP Wapper 与防火墙的功能最为相似,且不必实际用到大型的封包过滤.有上传文件,改变使用权限,和设定 GGI 程式等功能.安装了本软体之後,在控制上很明显的就可以轻易的作到,另一方面,也能够记录连接时的重要记录.在系统发生「不幸」时,那我们就可以参考这些记录,对方「凶手」的追查上颇有帮助(当然,前提这些记录要不被破坏).

762 ☆SELINUX

关于 SELINUX 服务器的解说 ，可能老手或新手对 SELINUX 都有点麻烦，建议您关掉 SELINUX;

在/etc/selinux/config 配置文件如下;
/etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - SELinux is fully disabled.
SELINUX=Disabled #这样就把 SELINUX 服务器关掉了， 请重新启动系统;
# SELINUXTYPE= type of policy in use. Possible values are:
# targeted - Only targeted network daemons are protected.
# strict - Full SELinux protection.
SELINUXTYPE=targeted

763 ☆使用身份验证服务前配置

关掉包过滤:在着手试验之前需要确认包过滤没有被激活，默认情况下 iptables 会调用 /etc/sysconfig/iptables 这个配置文件，删除或重命名这个文件，iptables 就会在下次启动时失效。
或者使用命令 chkconfig iptables off 也行。
如果想让 iptables 立刻失效可以用命令 service iptables stop.

764 ☆system-config-authentication

验证配置工具

765 ☆安装 NIS 服务器

ypserv，ypbind，和 yptools
nis 需要 portmap

766 ☆配置 NIS 服务器

编辑/etc/sysconfig/network
NISDOMAIN=域名

下次启动时才会起作用，设置了 NIS 域名之后不要重新启动，运行命令:
domainname <你们的 NIS 域名>

/var/yp/Makefile 文件 copy 一份作为备份

编辑 all 部分只包含 passwd 和 group:
all: passwd group

/var/yp/Makefile
makefile 中设定你服务器需要共享的信息
NOPUSH=true # 有辅服务器的时候为 false
MERGE_PASSWD=true # 是否合并 shadow 和 password 文件?
MERGE_GROUP=true # 是否合并 gshadow 和 group 文件?
MINUID=500 # 在 nis map 中最小的 uid
MINGID=500 # 在 nis map 中最小的 gid

打开 portmap 服务和 ypserv 服务
service portmap start
service ypserv start

确保 make 包在你的系统中安装
rpm -Uvh /mnt/rpms/RedHat/RPMS/make*

使用 ypinit 产生 NIS 数据库（maps），注意可能出现的错误信息
/usr/lib/yp/ypinit -m
（注意:你不用在列表中添加任何主机，只要按 < CTRL - D > ）

启动 NIS password 升级进程
service yppasswdd start

如果 ypinit 在第六步时没有错误，重新启动 ypserv 服务:
service ypserv restart

使用 ps auxf | grep yp 确定 ypserv 服务运行
如果有错误的话查看日志 /var/log/messages

767 ☆配置 NIS 辅服务器

在/etc/hosts 中增加 nis 主服务器

所有的辅服务器必须在主服务器的/var/yp/ypservers 中存在(/usr/lib/yp/ypinit -m 中会提示输入)

service portmap start
service ypserv start

/usr/lib/yp/ypinit -s <masterserver> 初始化从 nis 服务器

rpcinfo -p localhost 看 RPC 是否正常

768 ☆安装 NIS 客户端

portmap，ypbind，yp-tools 和 authconfig

769 ☆配置 NIS 客户端

NIS 仅仅提供验证信息，不提供客户端和服务器端的文件共享机制

确认客户端可以看到服务器上的 portmap 服务
rpcinfo -p 你们的 NIS 服务器

authconfig 工具配置你的客户端使用 NIS 进行身份验证
选定"Use NIS"，在"Domain:"后指定你的 NIS 域，在"Server:"后指定你的 NIS 服务器。

确认 authconfig 正确工作，当 authconfig 完成后，它会自动开启 ypbind 服务，是否有出错信息出现在控制台上或者 /var/log/messages 中

chkconfig ypbind on 也可以

domainname
返回 NIS 域名

测试你的 NIS 客户端，使用 root 用户登陆你的客户端，root 用户是客户端上的 root 还是 NIS 服务器上的？
NIS 服务器上的

测试 客户端----服务器的连接，使用:
ypcat passwd

这样会显示出 NIS 服务器上的 password 数据，（请记住，只有在服务器上/etc/passwd 文件中 UID 大于等于 500 的用户才会被放进数据库中）

使用 useradd 在客户端创建一个新的用户，然后在服务器端创建一个不同的用户，然后使用 passwd 设置他们的密码。
（在客户端）: useradd -u 1024 localguy
passwd localguy

（在服务器）: useradd -u 1025 nisuser
passwd nisuser

确认使用 localguy 能在本地登陆，nisuser 能在服务器上登陆。

然后使用 nisuser 帐号在客户端上登陆，应该是不可以的。

在服务器上的 /var/yp 目录，执行 make 命令(新加用户都后要这样)，当命令完成，再使用 nisuser 从客户端上登陆，这回应该成功了

使用 passwd 改变 nisuser 的密码，是否改变了服务器上的 /etc/passwd 和 /etc/shadow 文件？

NIS 服务器中的文件是否改变了呢？你可以使用如下命令测试:

ypcat passwd | grep nisuser

使用 localguy 登陆到客户端，是不是即时 ypbind 在运行仍然可以登陆？

当你使用 nisuser 登陆到客户端时，你的主目录是什么？

登录后没有目录
mkdir /home/panda
cp /etc/skel/.* /home/panda
chown panda.ldapusers /home/panda

/etc/sysconfig/network(可选步骤)
NISDIMAIN=域名

/etc/yp.conf
domain NISDOMAIN server HOSTNAME
NISDOMAIN 是 NIS 域名
HOSTNAME 是 NIS 的服务器

domain NISDOMAIN broadcast
在 NISDOMAIN 域上用广播

ypserver HOSTNAME
NIS 服务器的名字,必须在/etc/hosts 中

/etc/nsswitch.conf
passwd: nis file
shadow: nis file
group: nis file
先检查 nis 再检查本地的 file,可定义顺序

service ypbind start

/etc/resolv.conf
order nis,hists,bind

770 ☆NIS 客户命令

ypwhich
检查当前客户是用哪个 nis 服务器

下面都是更改 nis 服务器上的信息的
ypcat passwd
ypchfn 用户名
ypchsh 用户名
yppasswd 用户名
ypmatch 用户名 密码

yppush
把主 nis 服务器的信息拷贝到辅 nis 服务器
如果/var/yp/Makefile 中有"NOPUSH=false",则会自动拷贝

771 ☆LDAP 安装

nss_ldap
openldap_clinets

772 ☆配置 LDAP 客户端

authconfig
输入 ldap 服务器 ip
和基点 DN
dc=example,dc=com

LADP
/etc/ldap.conf
host 127.0.0.1
ldap 服务器的地址

base dc=example,dc=com
ldap 搜索数据库的名称

ssl no

有没有配置 SSL

pam_password md5
加密方法

/etc/nsswitch.conf
passwd: ldap
shadow: ldap
group: ldap

登录后没有目录
mkdir /home/panda
cp /etc/skel/.* /home/panda
chown panda.ldapusers /home/panda

web 上用 ldap 认证
Alias /mysecret "/usr/local/mysecret"
<Directory /usr/local/mysecret>
AuthType Basic
AuthName "Please Login:"
AuthLDAPURL "ldap://192.168.16.177/dc=example,dc=com"
require valid-user
</Directory>

773 ☆防火墙

防火墙是一套能够在两个或两个以上的网络之间，明显区隔出实体线路联机的软硬件设备组合。被区隔开来的网络，可以透过封包转送技术来相互通讯，透过防火墙的安全管理机制，可以决定哪些数据可以流通，哪些资料无法流通，藉此达到网络安全保护的目的。

防火墙产品可以概略归类为硬件式防火墙和软件式防火墙，但实际上无论是硬件式或软件式防火墙，它们都需要使用硬件来作为联机介接，也需要使用软件来设定安全政策，严格说两者间的差别并不太大。我们只能从使用的硬件与操作系统来加以区分，硬件式防火墙是使用专有的硬件，而软件式防火墙则使用一般的计算机硬件，硬件式防火墙使用专有的操作系统，而软件式防火墙则使用一般的操作系统。

防火墙依照其运作方式来分类，可以区分为封包过滤式防火墙 (Packet Filter) 、应用层网关式防火墙 (Application-Level Gateway，也有人把它称为 Proxy 防火墙)、电路层网关式防火墙 (Circuit-Level Gateway)。其中被广为采用的是封包过滤式防火墙，本文要介绍的 iptables 防火墙就是属于这一种。

封包过滤是最早被实作出来的防火墙技术，它是在 TCP/IP 四层架构下的 IP 层中运作。封包过滤器的功能主要是检查通过的每一个 IP 数据封包，如果其标头中所含的数据内容符合过滤条件的设定就进行进一步的处理，主要的处理方式包含:放行(accept)、丢弃(drop)或拒绝(reject)。要进行封包过滤，防火墙必须要能分析通过封包的来源 IP 与目的地 IP，还必须能检查封包类型、来源端口号与目的端口号、封包流向、封包进入防火墙的网卡接口、TCP 的联机状态等数据。

Linux 最早出现的防火墙软件称为 ipfw，ipfw 能透过 IP 封包标头的分析，分辨出封包的来源 IP 与目的地 IP、封包类型、来源端口号与目的端口号、封包流向、封包进入防火墙的网卡界面..... 等，并藉此分析结果来比对规则进行封包过滤，同时也支持 IP 伪装的功能，利用这个功能可以解决 IP 不足的问题，可惜这支程序缺乏弹性设计，无法自行建立规则组合(ruleset)作更精简的设定，同时也缺乏网址转译功能，无法应付越来越复杂的网络环境，而逐渐被淘汰。

取而代之的 ipchains，不但指令语法更容易理解，功能也较 ipfw 优越;ipchains 允许自订规则组合(ruleset)，称之为 user-define chains，透过这种设计，我们可以将彼此相关的规则组合在一起，在需要的时候跳到该组规则进行过滤，有效将规则的数量大幅缩减，以往 ipfw 仅能进行循序过滤，导致规则又臭又长的毛病，就不药而愈了。除了这个明显的好处以外，ipchains 并能结合本身的端口对应功能和 redir 程序的封包转送机制，模拟出网址转译的能力，而满足 NAT 的完整需求，堪称为一套成熟的防火墙作品。

防火墙软件的出现，确实曾经让骇客们晚上睡不着觉，因为防火墙的阻隔能够有效让内部网络不设防的单机不致于暴露在外，也能有效降低服务器的能见度，减少被攻击的机会，骇客过去所用的网络探测技术因此受到严格的挑战，越来越多的攻击对象躲藏在防火墙后方，让骇客难以接近，因此必须针对新的情势，研究出新的探测技术，藉以规避防火墙的检查，达到发现目标并进而攻击入侵的目的，新的技术非常多，本文并不拟进一步讨论，请自行参考 CERT 组织的技术文件，网址是 www.cert.org ，想看中文请连到 www.cert.org.tw

iptables 作为 ipchains 的新一代继承人，当然也针对骇客不断推陈出新的探测技术拟出一些因应之道，那就是对封包的联机状态，作出更详细的分析，例如:是否为新联机或响应封包、是否为转向联机、联机是否失去响应，联机时间是否过长.....等等，透过这样的分析能对一些可能被骇客利用的弱点加以阻隔(请详见后文的说明)，另外也开发出真正的封包改写能力，不需要透过其它程序的协助来仿真网址转译，除此之外，iptables 也获得系统核心的直接支持，不需要像 ipchains 那样需要自行重新编译核心。

iptables 优越的性能使它取代了 ipchains，成为网络防火墙的主流，而 ipchains 并未被淘汰，目前 ipchains 已经转型成单机防火墙，在安装新版 Linux 时，会自动被安装启用，以保护单机上未被使用的通讯端口。

774 ☆system-config-securitylevel-tui

防火墙配置工具

775☆iptables 入门

iptables 中的指令，均需区分大小写。

ipchains 和 iptables 在语法上的主要的差异，注意如下：
1. 在 ipchains 中，诸如 input 链，是使用小写的 chains 名，在 iptables 中，要改用大写 INPUT。
2. 在 iptables 中，要指定规则是欲作用在那一个规则表上(使用 -t 来指定，如 -t nat)，若不指定，则预设是作用在 filter 这个表。
3. 在 ipchains 中， -i 是指介面(interface)，但在 iptables 中，-i 则是指进入的方向，且多了 -o，代表出去的方向。
4. 在 iptables 中，来源 port 要使用关键字 --sport 或 --source-port
5. 在 iptables 中，目的 port 要使用关键字 --dport 或 --destination-port
6. 在 iptables 中，"丢弃" 的处置动作，不再使用 DENY 这个 target，改用 DROP。
7. 在 ipchains 的记录档功能 -l，已改为目标 -j LOG，并可指定记录档的标题。
8. 在 ipchains 中的旗标 -y，在 iptables 中可用 --syn 或 --tcp-flag SYN,ACK,FIN SYN
9. 在 iptables 中，imcp messages 型态，要加上关键字 --icmp-type，如：
iptables -A OUTPUT -o eth0 -p icmp -s $FW_IP --icmp-type 8 -d any/0 -j ACCEPT

iptables 使用时的样板

观察目前的设定
iptables -L -n
-n 是以数字打印端口
iptablse -t nat -L -n
指定 nat 表

定义变数
FW_IP="163.26.197.8"

打开核心 forward 功能
echo "1" > /proc/sys/net/ipv4/ip_forward
用 sysctl 打开 net.ipv4.ip_forward = 1 也可以

清除所有的规则
一开始要先清除所有的规则，重新开始，以免旧有的规则影响新的设定。
清除预设表 filter 中，所有规则链中的规则
iptables -F
清除预设表 filter 中，使用者自订链中的规则
iptables -X

清除 mangle 表中，所有规则链中的规则
iptables -t mangle -F

```
# 清除 mangle 表中，使用者自订链中的规则
iptables -t mangle -X

# 清除 nat 表中，所有规则链中的规则
iptables -t nat -F
# 清除 nat 表中，使用者自订链中的规则
iptables -t nat -X
```

选定预设的政策
接着，要选定各个不同的规则链，预设的政策为何。

预设全部丢弃：

```
# 设定 filter table 的预设政策
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

或者预设全部接受：

```
# 设定 filter table 的预设政策
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

各个规则链的预设政策可独立自主的设定，不必受其它链的影响。

776☆iptables

iptables 防火墙的指令非常类似于 ipchains，使用过 ipchains 的人应该很容易上手，但是 iptables 的机制与 ipchains 有很大的不同，使用 ipchains 的概念来设定规则，将会使防火墙无法正常运作。ipchains 跟 iptables 最大的不同在于对 INPUT、FORWARD 、OUTPUT 三个网络函式的定义不同，这三个网络函式是 TCP/IP 驱动程序的一部分，结构如下图所示，是介于网卡驱动程序和应用程序的中间，Linux 核心预设会启用 INPUT、OUTPUT 和 LOOPBACK，而 FORWARD 函式则必须自行启用，可以使用下面指令，或直接修改 /etc/sysconfig/network 组态档:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

- IP INPUT:所有封包都由 IP INPUT 函式负责处理，所以设定过滤规则时，几乎都是设定在 INPUT 规则链上。
- IP FORWARD:目的 IP 非本机的 IP，这些封包需要进一步作转送处理，此函式用来处理 IP 伪装和 Port 转送。 所有转送封包都在这里处理，这部分的过滤规则最复杂。
- IP OUTPUT:所有流出的封包都由这个函式处理，通常不需设定任何规则。

iptables 除了上述三支函式以外，还使用两个新的函式:Prerouting、Postrouting。

• PREROUTING:需要转送处理的封包由此函式负责处理，此函式用来做目的地 IP 的转译动作(DNAT)。
• POSTROUTING:转送封包送出之前，先透过这个函式进行来源 IP 的转译动作(SNAT)。

iptables 和 ipchains 都可以自行定义规则群组(rule-set)，规则群组被称为规则链(chains)，前面所描述的函式，也都有相对应的规则链(INPUT、FORWARD、OUTPUT、Prerouting、Postrouting)，为了有别于自行定义的规则链，这些规则链我们就称为内建规则链

777 ☆iptables 启动

service iptables restart

778 ☆iptables

iptables [-t 表] 命令 [匹配] [操作]

语法:
iptables [-t table] command [match] [-j target/jump]

779 ☆iptables 表选项

-t 参数用来指定规则表，内建的规则表有三个，分别是:nat、mangle 和 filter，当未指定规则表时，则一律视为是 filter。各个规则表的功能如下

nat 此规则表拥有 PREROUTING、POSTROUTING 和 OUTPUT 三个规则链，主要功能为进行一对一、一对多、多对多等网址转译工作(SNAT、DNAT)，由于转译工作的特性，需进行目的地网址转译的封包，就不需要进行来源网址转译，反之亦然，因此为了提升改写封包的效率，在防火墙运作时，每个封包只会经过这个规则表一次。如果我们把封包过滤的规则定义在这个数据表里，将会造成无法对同一封包进行多次比对，因此这个规则表除了作网址转译外，请不要做其它用途。

mangle 此规则表拥有 PREROUTING、POSTROUTING、INPUT、FORWARD 和 OUTPUT 五个规则链。
除了进行网址转译工作会改写封包外，在某些特殊应用可能也必须去改写封包(TTL、TOS)或者是设定 MARK(将封包作记号，以便进行后续的过滤)，这时就必须将这些工作定义在 mangle 规则表中，由于使用率不高，我们不打算在这里讨论 mangle 的用法。

filter 这个规则表是预设规则表，拥有 INPUT、FORWARD 和 OUTPUT 三个规则链，这个规则表顾名思义是用来进行封包过滤的处理动作(例如:DROP、LOG、ACCEPT 或 REJECT)，我们会将基本规则都建立在此规则表中。

780 ☆iptable 命令选项

常用命令列表:

下面的命令后面跟规则链名,可以是预设的,也可以是自定义的

命令 -L, --list
范例 iptables -L INPUT
说明 列出某规则链中的所有规则。

命令 -A, --append
范例 iptables -A INPUT ...
说明 新增规则到某个规则链中，该规则将会成为规则链中的最后一条规则。

命令 -I, --insert
范例 iptables -I INPUT 1 --dport 80 -j ACCEPT
说明 插入一条规则，原本该位置上的规则将会往后移动一个顺位。没有指定规则编号则在第一条前插入

命令 -D, --delete
范例 iptables -D INPUT --dport 80 -j DROP
iptables -D INPUT 1
说明 从某个规则链中删除一条规则，可以输入完整规则，或直接指定规则编号加以删除。

命令 -R, --replace
范例 iptables -R INPUT 1 -s 192.168.152.128 -j DROP
说明 取代现行规则，规则被取代后并不会改变顺序。必须指定规则编号

命令 -P, --policy
范例 iptables -P INPUT DROP
说明 定义过滤政策。 也就是未符合过滤条件之封包，预设的处理方式。

命令 -F, --flush
范例 iptables -F INPUT
说明 删除某规则链中的所有规则。

命令 -Z, --zero
范例 iptables -Z INPUT
说明 将封包计数器归零。封包计数器是用来计算同一封包出现次数，是过滤阻断式攻击不可或缺的工具。

命令 -N, --new-chain
范例 iptables -N allowed
说明 定义新的规则链。

命令 -E, --rename-chain

范例 iptables -E allowed disallowed
说明 修改某自订规则链的名称。

命令 -X, --delete-chain
范例 iptables -X allowed
说明 删除某个规则链。

781 ☆iptables 匹配选项

常用封包比对参数:

参数 -p, --protocol
范例 iptables -A INPUT -p tcp
说明 比对通讯协议类型是否相符，可以使用 ！运算子进行反向比对，例如:-p ! tcp ，意思是指除 tcp 以外的其它类型，包含 udp、icmp ...等。如果要比对所有类型，则可以使用 all 关键词，例如:-p all。

参数 -s, --src, --source
范例 iptables -A INPUT -s 192.169.152.129
说明 用来比对封包的来源 IP，可以比对单机或网络，比对网络时请用数字来表示屏蔽，例如:-s 192.168.0.0/24，比对 IP 时也可以使用 ！运算子进行反向比对，例如:-s ! 192.168.0.0/24。any/0 也可以

参数 -d, --dst, --destination
范例 iptables -A INPUT -d 192.169.152.129
说明 用来比对封包的目的地 IP，设定方式同上。可以为网址,www....

参数 -i, --in-interface
范例 iptables -A INPUT -i eth0
说明 用来比对封包是从哪片网卡进入，可以使用通配字符 + 来做大范围比对，例如:-i eth+ 表示所有的 ethernet 网卡，也可以使用 ！运算子进行反向比对，例如:-i ! eth0。

参数 -o, --out-interface
范例 iptables -A FORWARD -o eth0
说明 用来比对封包要从哪片网卡送出，设定方式同上。

参数 --sport, --source-port
范例 iptables -A INPUT -p tcp --sport 22
说明 用来比对封包的来源端口端口号，可以比对单一端口号，或是一个范围，例如:--sport 22:80，表示从 22 到 80 端口号之间都算是符合条件，如果要比对不连续的多个端口号，则必须使用 --multiport 参数，详见后文。比对端口号时，可以使用 ！运算子进行反向比对。

参数 --dport, --destination-port
范例 iptables -A INPUT -p tcp --dport 22
说明 用来比对封包的目的地端口号，设定方式同上。

参数 --tcp-flags
范例 iptables -p tcp --tcp-flags SYN,FIN,ACK SYN
说明 比对 TCP 封包的状态旗号，参数分为两个部分，第一个部分列举出想比对的旗号，第二部分则列举前述旗号中哪些有被设定，未被列举的旗号必须是空的。TCP 状态旗号包括:SYN(同步)、ACK(应答)、FIN(结束)、RST(重设)、URG(紧急)、PSH(强迫推送)等均可使用于参数中，除此之外还可以使用关键词 ALL 和 NONE 进行比对。比对旗号时，可以使用 ！运算子进行反向比对。

参数 --syn
范例 iptables -p tcp --syn
说明 用来比对是否为要求联机之 TCP 封包，与 iptables -p tcp --tcp-flags SYN,FIN,ACK SYN 的作用完全相同，如果使用 ！运算子，可用来比对非要求联机封包。

参数 -m multiport --source-port
范例 iptables -A INPUT -p tcp -m multiport --source-port 22,53,80,110
说明 用来比对不连续的多个来源端口号，一次最多可以比对 15 个端口，可以使用 ！运算子进行反向比对。

参数 -m multiport --destination-port
范例 iptables -A INPUT -p tcp -m multiport --destination-port 22,53,80,110
说明 用来比对不连续的多个目的地端口号，设定方式同上。

参数 -m multiport --port
范例 iptables -A INPUT -p tcp -m multiport --port 22,53,80,110
说明 这个参数比较特殊，用来比对来源端口号和目的端口号相同的封包，设定方式同上。注意:在本范例中，如果来源端口号为 80 但目的地端口号为 110，这种封包并不算符合条件。

参数 --icmp-type
范例 iptables -A INPUT -p icmp --icmp-type 8
说明 用来比对 ICMP 的类型编号，可以使用代码或数字编号来进行比对。请打 iptables -p icmp --help 来查看有哪些代码可以用。

参数 -m limit --limit
范例 iptables -A INPUT -m limit --limit 3/hour
说明 用来比对某段时间内封包的平均流量，上面的例子是用来比对:每小时平均流量是否超过一次 3 个封包。 除了每小时平均一次外，也可以每秒钟、每分钟或每天平均一次，默认值为每小时平均一次，参数如后: /second、 /minute、/day。 除了进行封包数量的比对外，设定这个参数也会在条件达成时，暂停封包的比对动作，以避免因骇客使用洪水攻击法，导致服务被阻断。

参数 --limit-burst
范例 iptables -A INPUT -m limit --limit-burst 5
说明 用来比对瞬间大量封包的数量，上面的例子是用来比对一次同时涌入的封包是否超过 5 个(这是默认值)，超过此上限的封包将被直接丢弃。使用效果同上。

参数 -m mac --mac-source
范例 iptables -A INPUT -m mac --mac-source 00:00:00:00:00:01
说明 用来比对封包来源网络接口的硬件地址，这个参数不能用在 OUTPUT 和 Postrouting 规则链上，这是因为封包要送出到网卡后，才能由网卡驱动程序透过 ARP 通讯协议查出目的地的 MAC 地址，所以 iptables 在进行封包比对时，并不知道封包会送到哪个网络接口去。

参数 --mark
范例 iptables -t mangle -A INPUT -m mark --mark 1
说明 用来比对封包是否被表示某个号码，当封包被比对成功时，我们可以透过 MARK 处理动作，将该封包标示一个号码，号码最大不可以超过 4294967296。

参数 -m owner --uid-owner
范例 iptables -A OUTPUT -m owner --uid-owner 500
说明 用来比对来自本机的封包，是否为某特定使用者所产生的，这样可以避免服务器使用 root 或其它身分将敏感数据传送出去，可以降低系统被骇的损失。可惜这个功能无法比对出来自其它主机的封包。

参数 -m owner --gid-owner
范例 iptables -A OUTPUT -m owner --gid-owner 0
说明 用来比对来自本机的封包，是否为某特定使用者群组所产生的，使用时机同上。

参数 -m owner --pid-owner
范例 iptables -A OUTPUT -m owner --pid-owner 78
说明 用来比对来自本机的封包，是否为某特定行程所产生的，使用时机同上。

参数 -m owner --sid-owner
范例 iptables -A OUTPUT -m owner --sid-owner 100
说明 用来比对来自本机的封包，是否为某特定联机(Session ID)的响应封包，使用时机同上。

参数 -m state --state
范例 iptables -A INPUT -m state --state RELATED,ESTABLISHED
说明 用来比对联机状态，联机状态共有四种:INVALID、ESTABLISHED、NEW 和 RELATED。
INVALID 表示该封包的联机编号(Session ID)无法辨识或编号不正确。
ESTABLISHED 表示该封包属于某个已经建立的联机。
NEW 表示该封包想要起始一个联机(重设联机或将联机重导向)。
RELATED 表示该封包是属于某个已经建立的联机，所建立的新联机。例如:FTP-DATA 联机必定是源自某个 FTP 联机。

782 ☆iptables 动作选项

常用的处理动作:

-j 参数用来指定要进行的处理动作，常用的处理动作包括:ACCEPT、REJECT、DROP、REDIRECT、MASQUERADE、LOG、DNAT、SNAT、MIRROR、QUEUE、RETURN、MARK，分别说明如下:

ACCEPT 将封包放行，进行完此处理动作后，将不再比对其它规则，直接跳往下一个规则链(nat:postrouting)。

REJECT 拦阻该封包，并传送封包通知对方，可以传送的封包有几个选择:ICMP port-unreachable、ICMP echo-reply 或是 tcp-reset(这个封包会要求对方关闭联机)，进行完此处理动作后，将不再比对其它规则，直接 中断过滤程序。 范例如下:
iptables -A FORWARD -p TCP --dport 22 -j REJECT --reject-with tcp-reset

DROP 丢弃封包不予处理，进行完此处理动作后，将不再比对其它规则，直接中断过滤程序。

REDIRECT 将封包重新导向到另一个端口(DNAT)，进行完此处理动作后，将会继续比对其它规则。 这个功能可以用来实作通透式 porxy 或用来保护 web 服务器。例如:iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080

MASQUERADE 改写封包来源 IP 为防火墙 NIC IP，可以指定 port 对应的范围，进行完此处理动作后，直接跳往下一个规则链(mangle:postrouting)。这个功能与 SNAT 略有不同，当进行 IP 伪装时，不需指定要伪装成哪个 IP，IP 会从网卡直接读取，当使用拨接连线时，IP 通常是由 ISP 公司的 DHCP 服务器指派的，这个时候 MASQUERADE 特别有用。范例如下:
iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000

LOG 将封包相关讯息纪录在 /var/log 中，详细位置请查阅 /etc/syslog.conf 组态档，进行完此处理动作后，将会继续比对其它规则。例如:
iptables -A INPUT -p tcp -j LOG --log-prefix "INPUT packets"

SNAT 改写封包来源 IP 为某特定 IP 或 IP 范围，可以指定 port 对应的范围，进行完此处理动作后，将直接跳往下一个规则链(mangle:postrouting)。范例如下:
iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to-source 194.236.50.155-194.236.50.160:1024-32000

DNAT 改写封包目的地 IP 为某特定 IP 或 IP 范围，可以指定 port 对应的范围，进行完此处理动作后，将会直接跳往下一个规则链(filter:input 或 filter:forward)。范例如下:

iptables -t nat -A PREROUTING -p tcp -d 15.45.23.67 --dport 80 -j DNAT --to-destination 192.169.152.129-192.169.152.130:80-100

MIRROR 镜射封包，也就是将来源 IP 与目的地 IP 对调后，将封包送回，进行完此处理动作后，将会中断过滤程序。

QUEUE 中断过滤程序，将封包放入队列，交给其它程序处理。透过自行开发的处理程序，可以进行其它应用，例如:计算联机费用.....等。

RETURN 结束在目前规则链中的过滤程序，返回主规则链继续过滤，如果把自订规则链看成是一个子程序，那么这个动作，就相当于提早结束子程序并返回到主程序中。

MARK 将封包标上某个代号，以便提供作为后续过滤的条件判断依据，进行完此处理动作后，将会继续比对其它规则。

范例如下:
iptables -t mangle -A PREROUTING -p tcp --dport 22 -j MARK --set-mark 2

783 ☆iptables 中文 man 文档

总览
用 iptables -ADC 来指定链的规则，-A 添加 -D 删除 -C 修改

iptables - [RI] chain rule num rule-specification[option]
用 iptables - RI 通过规则的顺序指定

iptables -D chain rule num[option]
删除指定规则

iptables -[LFZ] [chain][option]
用 iptables -LFZ 链名 [选项]

iptables -[NX] chain
用 -NX 指定链

iptables -P chain target[options]
指定链的默认目标

iptables -E old-chain-name new-chain-name
-E 旧的链名 新的链名
用新的链名取代旧的链名
说明

Iptalbes 是用来设置、维护和检查 Linux 内核的 IP 包过滤规则的。
可以定义不同的表，每个表都包含几个内部的链，也能包含用户定义的链。每个链都是一个规则列表，对对应的包进行匹配：每条规则指定应当如何处理与之相匹配的包。这被称作'target'（目标），也可以跳向同一个表内的用户定义的链。

TARGETS
防火墙的规则指定所检查包的特征，和目标。如果包不匹配，将送往该链中下一条规则检查;如果匹配,那么下一条规则由目标值确定.该目标值可以是用户定义的链名,或是某个专用值,如 ACCEPT[通过], DROP[删除], QUEUE[排队], 或者 RETURN[返回]。
ACCEPT 表示让这个包通过。DROP 表示将这个包丢弃。QUEUE 表示把这个包传递到用户空间。RETURN 表示停止这条链的匹配，到前一个链的规则重新开始。如果到达了一个内建的链(的末端)，或者遇到内建链的规则是 RETURN，包的命运将由链准则指定的目标决定。

TABLES
当前有三个表（哪个表是当前表取决于内核配置选项和当前模块)。
-t table
这个选项指定命令要操作的匹配包的表。如果内核被配置为自动加载模块，这时若模块没有加载，(系统)将尝试(为该表)加载适合的模块。这些表如下：filter,这是默认的表，包含了内建的链 INPUT（处理进入的包）、FORWORD（处理通过的包）和 OUTPUT（处理本地生成的包）。nat,这个表被查询时表示遇到了产生新的连接的包,由三个内建的链构成：PREROUTING (修改到来的包)、OUTPUT（修改路由之前本地的包）、POSTROUTING（修改准备出去的包）。mangle 这个表用来对指定的包进行修改。它有两个内建规则：PREROUTING（修改路由之前进入的包）和 OUTPUT（修改路由之前本地的包）。

OPTIONS
这些可被 iptables 识别的选项可以区分不同的种类。

COMMANDS
这些选项指定执行明确的动作：若指令行下没有其他规定,该行只能指定一个选项.对于长格式的命令和选项名,所用字母长度只要保证 iptables 能从其他选项中区分出该指令就行了。

-A -append
在所选择的链末添加一条或更多规则。当源（地址）或者/与 目的（地址）转换为多个地址时，这条规则会加到所有可能的地址(组合)后面。

-D -delete
从所选链中删除一条或更多规则。这条命令可以有两种方法：可以把被删除规则指定为链中的序号(第一条序号为 1),或者指定为要匹配的规则。

-R -replace
从选中的链中取代一条规则。如果源（地址）或者/与 目的（地址）被转换为多地址，该命令会失败。规则序号从 1 开始。

-I -insert
根据给出的规则序号向所选链中插入一条或更多规则。所以，如果规则序号为 1，规则会被插入链的头部。这也是不指定规则序号时的默认方式。

-L -list
显示所选链的所有规则。如果没有选择链，所有链将被显示。也可以和 z 选项一起使用，这时链会被自动列出和归零。精确输出受其它所给参数影响。

-F -flush
清空所选链。这等于把所有规则一个个的删除。

--Z -zero
把所有链的包及字节的计数器清空。它可以和 -L 配合使用，在清空前察看计数器，请参见前文。

-N -new-chain
根据给出的名称建立一个新的用户定义链。这必须保证没有同名的链存在。

-X -delete-chain
删除指定的用户自定义链。这个链必须没有被引用，如果被引用，在删除之前你必须删除或者替换与之有关的规则。如果没有给出参数，这条命令将试着删除每个非内建的链。

-P -policy
设置链的目标规则。

-E -rename-chain
根据用户给出的名字对指定链进行重命名，这仅仅是修饰，对整个表的结构没有影响。TARGETS 参数给出一个合法的目标。只有非用户自定义链可以使用规则，而且内建链和用户自定义链都不能是规则的目标。

-h Help.
帮助。给出当前命令语法非常简短的说明。

PARAMETERS
参数
以下参数构成规则详述，如用于 add、delete、replace、append 和 check 命令。

-p -protocal [!]protocol
规则或者包检查(待检查包)的协议。指定协议可以是 tcp、udp、icmp 中的一个或者全部，也可以是数值，代表这些协议中的某一个。当然也可以使用在/etc/protocols 中定义的协议名。在协议名前加上"!"表示相反的规则。数字 0 相当于所有 all。Protocol all 会匹配所有协议，而且这是缺省时的选项。在和 check 命令结合时，all 可以不被使用。

-s -source [!] address[/mask]
指定源地址，可以是主机名、网络名和清楚的 IP 地址。mask 说明可以是网络掩码或清楚的数字，在网络掩码的左边指定网络掩码左边"1"的个数，因此，mask 值为 24 等于 255.255.255.0。在指定地址前加上"!"说明指定了相反的地址段。标志 --src 是这个选项的简写。

-d --destination [!] address[/mask]
指定目标地址，要获取详细说明请参见 -s 标志的说明。标志 --dst 是这个选项的简写。

-j --jump target
-j 目标跳转
指定规则的目标;也就是说，如果包匹配应当做什么。目标可以是用户自定义链（不是这条规则所在的），某个会立即决定包的命运的专用内建目标，或者一个扩展（参见下面的 EXTENSIONS）。如果规则的这个选项被忽略，那么匹配的过程不会对包产生影响，不过规则的计数器会增加。

-i -in-interface [!] [name]
i -进入的（网络）接口 [!][名称]
这是包经由该接口接收的可选的入口名称，包通过该接口接收（在链 INPUT、FORWORD 和 PREROUTING 中进入的包）。当在接口名前使用"!"说明后，指的是相反的名称。如果接口名后面加上"+"，则所有以此接口名开头的接口都会被匹配。如果这个选项被忽略，会假设为"+"，那么将匹配任意接口。

-o --out-interface [!][name]
-o --输出接口[名称]
这是包经由该接口送出的可选的出口名称，包通过该口输出（在链 FORWARD、OUTPUT 和 POSTROUTING 中送出的包）。当在接口名前使用"!"说明后，指的是相反的名称。如果接口名后面加上"+"，则所有以此接口名开头的接口都会被匹配。如果这个选项被忽略，会假设为"+"，那么将匹配所有任意接口。

[!] -f, --fragment
[!] -f --分片
这意味着在分片的包中，规则只询问第二及以后的片。自那以后由于无法判断这种把包的源端口或目标端口（或者是 ICMP 类型的），这类包将不能匹配任何指定对他们进行匹配的规则。如果"!"说明用在了"-f"标志之前，表示相反的意思。

OTHER OPTIONS
其他选项
还可以指定下列附加选项：

-v --verbose
-v --详细
详细输出。这个选项让 list 命令显示接口地址、规则选项（如果有）和 TOS（Type of Service）掩码。包和字节计数器也将被显示，分别用 K、M、G(前缀)表示 1000、1,000,000 和 1,000,000,000

倍（不过请参看-x 标志改变它），对于添加,插入,删除和替换命令，这会使一个或多个规则的相关详细信息被打印。

-n --numeric
-n --数字
数字输出。IP 地址和端口会以数字的形式打印。默认情况下，程序试显示主机名、网络名或者服务（只要可用）。

-x -exact
-x -精确
扩展数字。显示包和字节计数器的精确值，代替用 K,M,G 表示的约数。这个选项仅能用于 -L 命令。

--line-numbers
当列表显示规则时，在每个规则的前面加上行号，与该规则在链中的位置相对应。

MATCH EXTENSIONS
对应的扩展
iptables 能够使用一些与模块匹配的扩展包。以下就是含于基本包内的扩展包，而且他们大多数都可以通过在前面加上!来表示相反的意思。

tcp
当 --protocol tcp 被指定,且其他匹配的扩展未被指定时,这些扩展被装载。它提供以下选项：

--source-port [!] [port[:port]]
源端口或端口范围指定。这可以是服务名或端口号。使用格式端口：端口也可以指定包含的（端口）范围。如果首端口号被忽略，默认是"0"，如果末端口号被忽略，默认是"65535"，如果第二个端口号大于第一个，那么它们会被交换。这个选项可以使用 --sport 的别名。

--destionation-port [!] [port:[port]]
目标端口或端口范围指定。这个选项可以使用 --dport 别名来代替。

--tcp-flags [!] mask comp
匹配指定的 TCP 标记。第一个参数是我们要检查的标记，一个用逗号分开的列表，第二个参数是用逗号分开的标记表,是必须被设置的。标记如下：SYN ACK FIN RST URG PSH ALL NONE。因此这条命令：iptables -A FORWARD -p tcp --tcp-flags SYN, ACK, FIN, RST SYN 只匹配那些 SYN 标记被设置而 ACK、FIN 和 RST 标记没有设置的包。

[!] --syn
只匹配那些设置了 SYN 位而清除了 ACK 和 FIN 位的 TCP 包。这些包用于 TCP 连接初始化时发出请求;例如，大量的这种包进入一个接口发生堵塞时会阻止进入的 TCP 连接，而出去的 TCP 连

接不会受到影响。这等于 --tcp-flags SYN, RST, ACK SYN。如果"--syn"前面有"!"标记，表示相反的意思。

--tcp-option [!] number
匹配设置了 TCP 选项的。

udp
当 protocol udp 被指定,且其他匹配的扩展未被指定时,这些扩展被装载,它提供以下选项：

--source-port [!] [port:[port]]
源端口或端口范围指定。详见 TCP 扩展的--source-port 选项说明。

--destination-port [!] [port:[port]]
目标端口或端口范围指定。详见 TCP 扩展的--destination-port 选项说明。

icmp
当 protocol icmp 被指定,且其他匹配的扩展未被指定时,该扩展被装载。它提供以下选项：
--icmp-type [!] typename
这个选项允许指定 ICMP 类型，可以是一个数值型的 ICMP 类型，或者是某个由命令 iptables -p icmp -h 所显示的 ICMP 类型名。

mac
--mac-source [!] address
匹配物理地址。必须是 XX:XX:XX:XX:XX 这样的格式。注意它只对来自以太设备并进入 PREROUTING、FORWORD 和 INPUT 链的包有效。

limit
这个模块匹配标志用一个标记桶过滤器一一定速度进行匹配,它和 LOG 目标结合使用来给出有限的登陆数.当达到这个极限值时,使用这个扩展包的规则将进行匹配.(除非使用了"!"标记)

--limit rate
最大平均匹配速率：可赋的值有'/second', '/minute', '/hour', or '/day'这样的单位，默认是 3/hour。

--limit-burst number
待匹配包初始个数的最大值:若前面指定的极限还没达到这个数值,则概数字加 1.默认值为 5

multiport
这个模块匹配一组源端口或目标端口,最多可以指定 15 个端口。只能和-p tcp 或者 -p udp 连着使用。

--source-port [port[, port]]
如果源端口是其中一个给定端口则匹配

--destination-port [port[, port]]
如果目标端口是其中一个给定端口则匹配

--port [port[, port]]
若源端口和目的端口相等并与某个给定端口相等,则匹配。

mark
这个模块和与 netfilter 过滤器标记字段匹配（就可以在下面设置为使用 MARK 标记）。

--mark value [/mask]
匹配那些无符号标记值的包（如果指定 mask，在比较之前会给掩码加上逻辑的标记）。

owner
此模块试为本地生成包匹配包创建者的不同特征。只能用于 OUTPUT 链，而且即使这样一些包（如 ICMP ping 应答）还可能没有所有者，因此永远不会匹配。

--uid-owner userid
如果给出有效的 user id，那么匹配它的进程产生的包。

--gid-owner groupid
如果给出有效的 group id，那么匹配它的进程产生的包。

--sid-owner seessionid
根据给出的会话组匹配该进程产生的包。

state
此模块，当与连接跟踪结合使用时，允许访问包的连接跟踪状态。

--state state
这里 state 是一个逗号分割的匹配连接状态列表。可能的状态是:INVALID 表示包是未知连接，ESTABLISHED 表示是双向传送的连接，NEW 表示包为新的连接，否则是非双向传送的，而 RELATED 表示包由新连接开始，但是和一个已存在的连接在一起，如 FTP 数据传送，或者一个 ICMP 错误。

unclean
此模块没有可选项，不过它试着匹配那些奇怪的、不常见的包。处在实验中。

tos
此模块匹配 IP 包首部的 8 位 tos（服务类型）字段（也就是说，包含在优先位中）。

--tos tos

这个参数可以是一个标准名称，（用 iptables -m tos -h 察看该列表），或者数值。

TARGET EXTENSIONS
iptables 可以使用扩展目标模块：以下都包含在标准版中。

LOG
为匹配的包开启内核记录。当在规则中设置了这一选项后，linux 内核会通过 printk()打印一些关于全部匹配包的信息（诸如 IP 包头字段等）。

--log-level level
记录级别（数字或参看 syslog.conf(5)）。

--log-prefix prefix
在纪录信息前加上特定的前缀：最多 14 个字母长，用来和记录中其他信息区别。

--log-tcp-sequence
记录 TCP 序列号。如果记录能被用户读取那么这将存在安全隐患。

--log-tcp-options
记录来自 TCP 包头部的选项。

--log-ip-options
记录来自 IP 包头部的选项。

MARK
用来设置包的 netfilter 标记值。只适用于 mangle 表。

--set-mark mark

REJECT
作为对匹配的包的响应，返回一个错误的包：其他情况下和 DROP 相同。

此目标只适用于 INPUT、FORWARD 和 OUTPUT 链，和调用这些链的用户自定义链。这几个选项控制返回的错误包的特性：

--reject-with type
Type 可以是 icmp-net-unreachable、icmp-host-unreachable、icmp-port-nreachable、icmp-proto-unreachable、icmp-net-prohibited 或者 icmp-host-prohibited，该类型会返回相应的 ICMP 错误信息（默认是 port-unreachable）。选项 echo-reply 也是允许的;它只能用于指定 ICMP ping 包的规则中，生成 ping 的回应。最后，选项 tcp-reset 可以用于在 INPUT 链中,或自 INPUT 链调用的规则，只匹配 TCP 协议：将回应一个 TCP RST 包。

TOS
用来设置 IP 包的首部八位 tos。只能用于 mangle 表。

--set-tos tos
你可以使用一个数值型的 TOS 值，或者用 iptables -j TOS -h 来查看有效 TOS 名列表。

MIRROR
这是一个试验示范目标，可用于转换 IP 首部字段中的源地址和目标地址，再传送该包,并只适用于 INPUT、FORWARD 和 OUTPUT 链，以及只调用它们的用户自定义链。

SNAT
这个目标只适用于 nat 表的 POSTROUTING 链。它规定修改包的源地址（此连接以后所有的包都会被影响），停止对规则的检查，它包含选项：

--to-source <ipaddr>[-<ipaddr>][:port-port]
可以指定一个单一的新的 IP 地址，一个 IP 地址范围，也可以附加一个端口范围（只能在指定-p tcp 或者-p udp 的规则里）。如果未指定端口范围，源端口中 512 以下的（端口）会被安置为其他的 512 以下的端口;512 到 1024 之间的端口会被安置为 1024 以下的，其他端口会被安置为 1024 或以上。如果可能，端口不会被修改。

--to-destiontion <ipaddr>[-<ipaddr>][:port-port]
可以指定一个单一的新的 IP 地址，一个 IP 地址范围，也可以附加一个端口范围（只能在指定-p tcp 或者-p udp 的规则里）。如果未指定端口范围，目标端口不会被修改。

MASQUERADE
只用于 nat 表的 POSTROUTING 链。只能用于动态获取 IP（拨号）连接：如果你拥有静态 IP 地址，你要用 SNAT。伪装相当于给包发出时所经过接口的 IP 地址设置一个映像，当接口关闭连接会终止。这是因为当下一次拨号时未必是相同的接口地址（以后所有建立的连接都将关闭）。它有一个选项：

--to-ports <port>[-port>]
指定使用的源端口范围，覆盖默认的 SNAT 源地址选择（见上面）。这个选项只适用于指定了-p tcp 或者-p udp 的规则。

REDIRECT
只适用于 nat 表的 PREROUTING 和 OUTPUT 链，和只调用它们的用户自定义链。它修改包的目标 IP 地址来发送包到机器自身（本地生成的包被安置为地址 127.0.0.1）。它包含一个选项：

--to-ports <port>[<port>]
指定使用的目的端口或端口范围：不指定的话，目标端口不会被修改。只能用于指定了-p tcp 或 -p udp 的规则。

DIAGNOSTICS
诊断
不同的错误信息会打印成标准错误：退出代码 0 表示正确。类似于不对的或者滥用的命令行参数错误会返回错误代码 2，其他错误返回代码为 1。

BUGS
臭虫
Check is not implemented (yet).
检查还未完成。

COMPATIBILITY WITH IPCHAINS
与 ipchains 的兼容性
iptables 和 Rusty Russell 的 ipchains 非常相似。主要区别是 INPUT 链只用于进入本地主机的包,而 OUTPUT 只用于自本地主机生成的包。因此每个包只经过三个链的一个;以前转发的包会经过所有三个链。其他主要区别是 -i 引用进入接口;-o 引用输出接口，两者都适用于进入 FORWARD 链的包。当和可选扩展模块一起使用默认过滤器表时，iptables 是一个纯粹的包过滤器。这能大大减少以前对 IP 伪装和包过滤结合使用的混淆，所以以下选项作了不同的处理：
-j MASQ
-M -S
-M -L
在 iptables 中有几个不同的链。

784 ☆定义默认策略

iptables [-t 表名] [-P] [链名] [动作]

iptables -P INPUT ACCEPT
filter 表的 input 链的默认策略为接收数据包

基本原则
先拒绝所有数据包,再允许需要数据包

filter 表的默认可为
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

785 ☆查看 iptables 规则

iptables [-t 表名] [-L] [链名]

iptables -t nat -L

看 nat 表所有链规则

786 ☆增加,插入,删除和替换规则

iptables [-t 表名] [-A|I|D|R] [链名] [规则编号] [-i|o 网卡名称] [-p 协议类型] [-s 源 ip|源子网] [--sport 源端口号] [-d 目标 ip|目标子网] [-dport 目标端口号] [-j 动作]

规则是按编号从前往后读,前面拒绝了,后面就无法接收

787 ☆清除规则和计数器

iptables [-t 表名] [-F|Z]

788 ☆保存恢复 iptables 规则

用上述方法所建立的规则会被保存到内核中，当重新引导系统时，会丢失这些规则。所以，如果您将没有错误的且有效的规则集添加到信息包过滤表，同时希望在重新引导之后再次使用这些规则，那么必须将该规则集保存在文件中。可以使用 iptables-save 命令来做到这一点：
$ iptables-save > iptables-script
现在，信息包过滤表中的所有规则都被保存在文件 iptables-script 中。无论何时再次引导系统，都可以使用 iptables-restore 命令将规则集从该脚本文件恢复到信息包过滤表，如下所示：
$ iptables-restore iptables-script

iptables-save > /etc/sysconfig/iptables
保存防火墙配置
或者
sevice iptables save

789 ☆iptables 实现 NAT

打开内核路由功能
echo "1" > /proc/sys/net/ipv4/ip_forward

实现 ip 伪装,输出到 ppp0
iptables -t nat -I POSTROUTING -o ppp0 -j MASQUERADE

NAT 客户端配置
NAT 客户端的默认网关为 NAT 服务器的内部网卡 ip 地址,DNS 为 ISP 提供的
修改/etc/sysconfig/network
GATEWAY=
修改/etc/resolv.conf
nameserver

上面的加到/etc/rc.d/rc.local 文件的末尾,可以自动启动

790 ☆创建一个简单的防火墙

你要建立一个防火墙保护你的主机不受可疑主机 192.168.0.254 的骚扰，可疑的主机不只这一个，你还要创建一个规则防止你的一个邻近的主机使用 ping-flooding（洪水 ping）攻击你的计算机。

删除所有已经存在的用户定义的 chains，重置所有 chains 上的默认规则，刷新所有规则:
iptables -F; iptables -X
for chain in INPUT FORWARD OUTPUT; do
iptables -p $chain ACCEPT
done
或者
service iptables stop

10. 阻止所有从邻近的主机（192.168.0.Y）的进来的连接:

iptables -A INPUT -s 192.168.0.Y -m state --state NEW -j DROP

这样还是允许你打开到他们系统的连接，但不是所有的

11. 限制从你的邻居（192.168.0.X）进来的 ICMP echo request（回应请求）包

iptables -A INPUT -s 192.168.0.X -p icmp --icmp-type echo-request \
-m limit --limit 6/minute --limit-burst 2 -j ACCEPT

iptables -A INPUT -s 192.168.0.X -p icmp --icmp-type echo-request \
-j DROP

12. 显示你的防火墙策略

iptables -nl

13. 测试你的防火墙配置
a) 你的邻居（192.168.0.Y）能连接到你的系统吗？你能 ping 通他吗？
b) 确认你的邻居（192.168.0.X）使用的不是你在上面第 2 步时设置的地址。
c) 你的邻居（192.168.0.X）能 ping 通你的系统吗？你能 ping 通他吗？

14. 保存你的防火墙设置:

iptables-save > /etc/sysconfig/iptables

或者
service iptables save

15. 配置你的系统重启后仍保留新的防火墙规则:

chkconfig --level 2345 iptables on

现在确认一下

chkconfig --list iptables

16. 重新启动确认你的策略仍在。

完成:
1. 你可以主动连接你的邻居（192.168.0.Y）
2. 所有的主机都可以主动连接你，除了你的邻居（192.168.0.Y）
3. 你的另一个邻居（192.168.0.X）不能用 ping-flood 攻击你的系统。

清理:
当你确信成功完成了实验，让你刚才创建的策略实效:

service iptables stop
chkconfig iptables off

791 ☆iptables 应用实例

```
#!/bin/sh
#
# 石牌国小防火墙设定指令稿
# 2002/8/27
# 设定者:李忠宪(修改自 iptables tutorial 1.1.11 by Oskar Andreasson )
# 原文件是依 DMZ 需求设计，已根据校园 NAT 网络之需求修改，其余改动部份包括:
# 新增通讯协议定义区块
# 新增执行时，自动清除已设定之规则
# 支援 FTP
# 修改所有规则，改采 multiport 方式以简化规则
# 原文件仅支持 IP 伪装(多对一对应)，已扩充为支持一对一对应及多对多对应
# 原文件仅支援 DNS 及 WEB，新增 ftp、mail、wam、PCAnywhere、ssh.....等多种服务器
# 修改若干规则设定上的小错误
#
# Copyright (C) 2001 Oskar Andreasson <bluefluxATkoffeinDOTnet>
#
```

```
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; version 2 of the License.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program or from the site that you downloaded it
# from; if not, write to the Free Software Foundation, Inc., 59 Temple
# Place, Suite 330, Boston, MA 02111-1307 USA
#
  ..##############################################################################
#
# 1. Configuration options.
#
# 1.0 Protocols Configuration.
#  定义会用到的通讯协议
HTTP="80"
HTTPS="443"
FTP="21"
FTP_DATA="20"
SMTP="25"
POP3="110"
IMAP="143"
SSH="22"
TELNET="23"
PCAW_TCP="5631"
PCAW_UDP="5632"
WEBMIN="10000"
WAM="12000"
DNS="53"

#
# 1.1 Internet Configuration.
#
#  定义 NIC IP 及 WAN 接口
INET_IP="163.21.xxx.253"
HTTP1_IP="163.21.xxx.2"
HTTP2_IP="163.21.xxx.4"
```

```
HTTP3_IP="163.21.xxx.9"
HTTP4_IP="163.21.xxx.6"
HTTP5_IP="163.21.xxx.7"
HTTP6_IP="163.21.xxx.10"
FTP1_IP="163.21.xxx.2"
FTP2_IP="163.21.xxx.6"
FTP3_IP="163.21.xxx.7"
MAIL1_IP="163.21.xxx.6"
MAIL2_IP="163.21.xxx.7"
PCAW1_IP="163.21.xxx.2"
PCAW2_IP="163.21.xxx.4"
WAM1_IP="163.21.xxx.6"
WAM2_IP="163.21.xxx.7"
DNS_IP="163.21.xxx.2"
IP_POOL="163.21.xxx.240-163.21.xxx.250"
INET_IFACE="eth1"

#
# 1.2 Local Area Network configuration.
#
# 定义 NAT IP 及 LAN 接口
LAN_IP="192.169.152.12853"
LAN_HTTP1_IP="192.169.152.128"
LAN_HTTP2_IP="192.168.1.4"
LAN_HTTP3_IP="192.168.1.9"
LAN_HTTP4_IP="192.168.1.6"
LAN_HTTP5_IP="192.168.1.7"
LAN_HTTP6_IP="192.168.1.53"
LAN_FTP1_IP="192.169.152.128"
LAN_FTP2_IP="192.168.1.6"
LAN_FTP3_IP="192.168.1.7"
LAN_MAIL1_IP="192.168.1.6"
LAN_MAIL2_IP="192.168.1.7"
LAN_PCAW1_IP="192.169.152.128"
LAN_PCAW2_IP="192.168.1.4"
LAN_WAM1_IP="192.168.1.6"
LAN_WAM2_IP="192.168.1.7"
LAN_DNS_IP="192.169.152.128"
LAN_IP_RANGE="192.168.0.0/16"
LAN_BROADCAST_ADDRESS="192.169.152.12855"
LAN_IFACE="eth0"
```

```
#
# 1.4 Localhost Configuration.
#
#  定义  Loopback IP  及接口
LO_IFACE="lo"
LO_IP="127.0.0.1"

#
# 1.5 IPTables Configuration.
#
#  设定  iptables  指令路径
IPTABLES="/sbin/iptables"

#
# 1.6 Other Configuration.
#
  ..#############################################################################
#
# 2. Module loading.
#
#
# Needed to initially load modules
#  整理核心支持模块之清单
/sbin/depmod -a

#
# 2.1 Required modules
#  加载会用到的模块
/sbin/modprobe ip_tables
/sbin/modprobe ip_conntrack
/sbin/modprobe iptable_filter
/sbin/modprobe iptable_mangle
/sbin/modprobe iptable_nat
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_limit
/sbin/modprobe ipt_state
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp

#
# 2.2 Non-Required modules
#  其余未使用之模块
```

```
#/sbin/modprobe ipt_owner
#/sbin/modprobe ipt_REJECT
#/sbin/modprobe ipt_MASQUERADE
#/sbin/modprobe ip_conntrack_irc
#/sbin/modprobe ip_nat_irc
  ..#############################################################################
#
# 3. /proc set up.
#
#
# 3.1 Required proc configuration
#  启动 Forward 接口
echo "1" > /proc/sys/net/ipv4/ip_forward

#
# 3.2 Non-Required proc configuration
#  其余未使用之接口
#echo "1" > /proc/sys/net/ipv4/conf/all/rp_filter
#echo "1" > /proc/sys/net/ipv4/conf/all/proxy_arp
#echo "1" > /proc/sys/net/ipv4/ip_dynaddr
  ..#############################################################################
#
# 4. rules set up.
#
  ..##
# 4.1 Filter table
#
# 4.1.0 Reset the default policies in the nat table.
#  清除所有已设定之规则，回复到不设防状态
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -t nat -P PREROUTING ACCEPT
$IPTABLES -t nat -P POSTROUTING ACCEPT
$IPTABLES -t nat -P OUTPUT ACCEPT
$IPTABLES -t mangle -P PREROUTING ACCEPT
$IPTABLES -t mangle -P OUTPUT ACCEPT
$IPTABLES -F
$IPTABLES -t nat -F
$IPTABLES -t mangle -F
$IPTABLES -X
$IPTABLES -t nat -X
```

```
$IPTABLES -t mangle -X

#
# 4.1.1 Set policies
#  定义安全政策为正面表列
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

#
# 4.1.2 Create userspecified chains
#
#
#  新增使用者自订规则链 bad_tcp_packets、 allowed 和 icmp_packets
$IPTABLES -N bad_tcp_packets
$IPTABLES -N allowed
$IPTABLES -N icmp_packets

#
# 4.1.3 Create content in userspecified chains
#
#
# bad_tcp_packets chain
# bad_tcp_packets 规则链的功能是:将要求重导向的联机记录起来，然后将封包丢弃(防止联机被绑架，但会档掉需要三方交谈的服务，例如:M$ Media Server)
$IPTABLES -A bad_tcp_packets -p tcp ! --syn -m state --state NEW -j LOG --log-level INFO
--log-prefix "New not syn:"
$IPTABLES -A bad_tcp_packets -p TCP ! --syn -m state --state NEW -j DROP

# allowed chain
# allowed 规则链的功能是:允许要求联机封包或响应封包进入，将其余封包丢弃
$IPTABLES -A allowed -p TCP --syn -j ACCEPT
$IPTABLES -A allowed -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A allowed -p TCP -j DROP

#
# ICMP rules
# icmp_packets 规则链的功能是:允许 ping 封包进入，将其余封包丢弃
$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 8 -j ACCEPT
$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 11 -j ACCEPT

#
```

```
# 4.1.4 INPUT chain(过滤要到达防火墙的封包)
#
#
# 进入防火墙主机的 TCP 封包必须先进行 bad_tcp_packets 过滤
$IPTABLES -A INPUT -p tcp -j bad_tcp_packets

# 从 WAN 进入防火墙主机的 ICMP 封包必须先进行 icmp_packets 过滤，这是为了避免骇客传送不完整的 IP 封包，系统会响应 ICMP 封包通知对方，导致主机位置被侦测出来
$IPTABLES -A INPUT -p ICMP -i $INET_IFACE -j icmp_packets

# 从 LAN 进入防火墙主机的全部 unicast 和 broadcast 封包，通通放行;额外检查目的地 IP 可以将 multicast 封包滤除
$IPTABLES -A INPUT -p ALL -i $LAN_IFACE -d $LAN_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LAN_IFACE -d $LAN_BROADCAST_ADDRESS -j ACCEPT

# 从 Loopback 接口进入防火墙主机的所有封包，检查是否来自本机，若是则放行;此规则去检查来源 IP ，似乎有些画蛇添足，因为只有来自本机的封包才有机会进入 Loopback 接口
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LO_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LAN_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $INET_IP -j ACCEPT

# 从 LAN 进入防火墙主机的 DHCP 封包，予以放行，只有当防火墙担任 DHCP 时才使用
#$IPTABLES -A INPUT -p UDP -i $LAN_IFACE --dport 67 --sport 68 -j ACCEPT

# 从 WAN 进入防火墙主机的所有封包，检查是否为响应封包，若是则予以放行
$IPTABLES -A INPUT -p ALL -d $INET_IP -m state --state ESTABLISHED,RELATED -j ACCEPT

# 限制过滤规则的比对频率为每分钟平均流量三个封包(超过上限的封包将暂停比对)，并将瞬间流量设定为一次最多处理三个封包(超过上限的封包将丢弃不予处理)，这类封包通常是骇客用来进行阻断式攻击
$IPTABLES -A INPUT -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level INFO --log-prefix
"IPT INPUT packet died: "

#
# 4.1.5 FORWARD chain(过滤要通过防火墙的封包)
#
#
# 通过防火墙的 TCP 封包必须先进行 bad_tcp_packets 过滤
$IPTABLES -A FORWARD -p TCP -j bad_tcp_packets

# 从 LAN 要到 WAN 的封包通通放行
$IPTABLES -A FORWARD -i $LAN_IFACE -o $INET_IFACE -j ACCEPT
```

```
# 从 WAN 要到 LAN 的封包仅放行回应封包
$IPTABLES -A FORWARD -i $INET_IFACE -o $LAN_IFACE -m state --state
ESTABLISHED,RELATED -j ACCEPT

# 允许来自 WAN 的 Ping 封包，递送到校内所有的服务器
$IPTABLES -A FORWARD -p ICMP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP1_IP -j
icmp_packets
$IPTABLES -A FORWARD -p ICMP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP2_IP -j
icmp_packets
$IPTABLES -A FORWARD -p ICMP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP3_IP -j
icmp_packets
$IPTABLES -A FORWARD -p ICMP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP4_IP -j
icmp_packets
$IPTABLES -A FORWARD -p ICMP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP5_IP -j
icmp_packets
$IPTABLES -A FORWARD -p ICMP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP6_IP -j
icmp_packets

# 允许来自 WAN 的 HTTP、HTTPS 封包，递送到校内所有的 WEB 服务器
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP1_IP -m
multiport --dport $HTTP,$HTTPS -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP2_IP -m
multiport --dport $HTTP,$HTTPS -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP3_IP -m
multiport --dport $HTTP,$HTTPS -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP4_IP -m
multiport --dport $HTTP,$HTTPS -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP5_IP -m
multiport --dport $HTTP,$HTTPS -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_HTTP6_IP -m
multiport --dport $HTTP,$HTTPS -j allowed

# 允许来自 WAN 的 FTP 封包，递送到校内所有的 FTP 服务器
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_FTP1_IP -m
multiport --dport $FTP,$FTP_DATA -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_FTP2_IP -m
multiport --dport $FTP,$FTP_DATA -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_FTP3_IP -m
multiport --dport $FTP,$FTP_DATA -j allowed

# 允许来自 WAN 的 SMTP、POP3、IMAP 封包，递送到校内所有的 MAIL 服务器
```

```
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_MAIL1_IP -m
multiport --dport $SMTP,$POP3,$IMAP -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_MAIL2_IP -m
multiport --dport $SMTP,$POP3,$IMAP -j allowed

# 允许来自 WAN 的 SSH、TELNET、WEBMIN、WAM 封包，递送到校内所有的 LINUX 服
务器
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_WAM1_IP -m
multiport --dport $SSH,$TELNET,$WEBMIN,$WAM -j allowed
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_WAM2_IP -m
multiport --dport $SSH,$TELNET,$WEBMIN,$WAM -j allowed

# 允许来自 WAN 的 PCanywhere 封包，递送到校内所有的 PCanywhere 服务器
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_PCAW1_IP --dport
$PCAW_TCP -j allowed
$IPTABLES -A FORWARD -p UDP -i $INET_IFACE -o $LAN_IFACE -d $LAN_PCAW1_IP --dport
$PCAW_UDP -j ACCEPT
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_PCAW2_IP --dport
$PCAW_TCP -j allowed
$IPTABLES -A FORWARD -p UDP -i $INET_IFACE -o $LAN_IFACE -d $LAN_PCAW2_IP --dport
$PCAW_UDP -j ACCEPT

# 允许来自 WAN 的 DNS 封包，递送到校内的 DNS 服务器
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $LAN_IFACE -d $LAN_DNS_IP --dport
$DNS -j allowed
$IPTABLES -A FORWARD -p UDP -i $INET_IFACE -o $LAN_IFACE -d $LAN_DNS_IP --dport
$DNS -j ACCEPT

# 限制过滤规则的比对频率为每分钟平均流量三个封包(超过上限的封包将暂停比对)，并将瞬间
流量设定为一次最多处理三个封包(超过上限的封包将丢弃不予处理)，这类封包通常是骇客用来
进行阻断式攻击
$IPTABLES -A FORWARD -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level DEBUG
--log-prefix "IPT FORWARD packet died: "

#
# 4.1.6 OUTPUT chain(过滤从防火墙送出的封包)
#
#
# 从防火墙送出的 TCP 封包必须先进行 bad_tcp_packets 过滤
$IPTABLES -A OUTPUT -p TCP -j bad_tcp_packets

# 从防火墙网卡送出的所有封包，通通放行
```

```
$IPTABLES -A OUTPUT -p ALL -s $LO_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $LAN_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $INET_IP -j ACCEPT

# 限制过滤规则的比对频率为每分钟平均流量三个封包(超过上限的封包将暂停比对)，并将瞬间流量设定为一次最多处理三个封包(超过上限的封包将丢弃不予处理)，这类封包通常是骇客用来进行阻断式攻击
$IPTABLES -A OUTPUT -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level DEBUG
--log-prefix "IPT OUTPUT packet died: "

  ..##
# 4.2 nat table
#
#
# 4.2.1 Set policies
#
#
# 4.2.2 Create user specified chains
#
#
# 4.2.3 Create content in user specified chains
#
#
# 4.2.4 PREROUTING chain(定义目的地地址转译)
#
# 从 WAN 要到校内服务器的封包，在封包过滤前先转译目的地 IP 为 NAT IP
$IPTABLES -t nat -A PREROUTING -d $HTTP1_IP -j DNAT --to-destination $LAN_HTTP1_IP
$IPTABLES -t nat -A PREROUTING -d $HTTP2_IP -j DNAT --to-destination $LAN_HTTP2_IP
$IPTABLES -t nat -A PREROUTING -d $HTTP3_IP -j DNAT --to-destination $LAN_HTTP3_IP
$IPTABLES -t nat -A PREROUTING -d $HTTP4_IP -j DNAT --to-destination $LAN_HTTP4_IP
$IPTABLES -t nat -A PREROUTING -d $HTTP5_IP -j DNAT --to-destination $LAN_HTTP5_IP
$IPTABLES -t nat -A PREROUTING -d $HTTP6_IP -j DNAT --to-destination $LAN_HTTP6_IP

#
# 4.2.5 POSTROUTING chain(定义来源地址转译)
#
# 从校内服务器要到 WAN 的封包，在送出之前先转译来源 IP 为 NIC IP，配合上面区块的设定，就可以做到一对一对应
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -s $LAN_HTTP1_IP -j SNAT --to-source
$HTTP1_IP
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -s $LAN_HTTP2_IP -j SNAT --to-source
$HTTP2_IP
```

```
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -s $LAN_HTTP3_IP -j SNAT --to-source
$HTTP3_IP
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -s $LAN_HTTP4_IP -j SNAT --to-source
$HTTP4_IP
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -s $LAN_HTTP5_IP -j SNAT --to-source
$HTTP5_IP
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -s $LAN_HTTP6_IP -j SNAT --to-source
$HTTP6_IP
# 从校内一般单机要到 WAN 的封包，在送出之前先转译来源 IP 为预设的 NIC IP，这就是多
对一对应，若指定成 IP 范围，就变成多对多对应，例如本范例即是如此
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j SNAT --to-source $IP_POOL

#
# 4.2.6 OUTPUT chain
#
  ..##
# 4.3 mangle table
#
#
# 4.3.1 Set policies
#
#
# 4.3.2 Create user specified chains
#
#
# 4.3.3 Create content in user specified chains
#
#
# 4.3.4 PREROUTING chain
#
#
# 4.3.5 INPUT chain
#
#
# 4.3.6 FORWARD chain
#
#
# 4.3.7 OUTPUT chain
#
#
# 4.3.8 POSTROUTING chain
#
```

柒、Log 分析

分析 iptables 防火墙 Log 的免费软件相当多，底下仅介绍 iptables_logger_v0.3，这个软件提供一个 perl 程序，可以读取系统 LOG，并将数据写入 MySql 数据库，然后还提供 php 程序，可以从数据库读取数据，整理成网页提供浏览，因此要安装此分析软件，必须先安装 perl、php、mysql 和 apache，有关这些套件的安装在这里不再介绍，请自行参考相关文件，或参加 Linux 进阶班课程。你可以从这里取得 iptables_logger_v0.3 程序，其安装程序如下:

安装数据表:

这个套件解压缩后，可以看到有一个数据夹叫做 sql，数据夹内有一个 sql 的指令稿叫做 db.sql，这个指令稿是用来建立摆放联机纪录所需的数据表，请利用以下指令来安装，如果您还不熟悉 mysql 的指令，请自行阅读 man mysql 文件。

root@firewall sql# mysql -u root -p(以 root 身分登入 MySql 主控台)

mysql> create database iptables;(建立一个数据库叫做 iptables，数据库也可以自行命名，但是要记得修改相关程序)

mysql> grant create,select,insert on iptables.* to iptables_admin@localhost identified by 'xx';(将 iptables 数据库新建、读取和写入权限授权给 iptables_admin 这个账号，并限制只能从本机联机，密码为 xx，请自行修改上述指令中之账号与密码)

mysql> grant create,select on iptables.* to iptables_user@localhost identified by 'xx';(将 iptables 数据库读取权限授权给 iptables_user 这个账号，并限制只能从本机联机，密码为 xx，请自行修改上述指令中之账号与密码)

root@firewall sql# cat db.sql | mysql -u iptables_admin -p iptables(以 iptables_admin 身分来执行 db.sql，如果你改了数据库的名字，请记得修改 db.sql)

修改 iptables 指令稿:

由于这支 Log 分析程序是以读取系统 LOG 加以分析后才汇入到数据库的方法，来处理联机纪录，并非使用 ULOG 直接由 iptables 将纪录写入 mysql 数据库，感觉上效能比较差，但在 ULOG 机制尚未被实体化之前，这也不失为一个好的解决方案。

原则上只要在 iptables 指令稿中有产生 LOG 的动作，这些信息就会被分析汇入到数据表(由 feed_db.pl 负责这个工作)，特别要注意的是 LOG 产生时会加入一个标头(prefix)，程序是透过标头来分析这笔 LOG 的意义，请将 LOG 标头取名为 IPTABLES DROP 或 IPTABLES ACCEPT，以方便事后的统计。范例如下:

首先建立一个新的规则链 LOG_DROP，这个规则链用来将要丢弃的封包先 LOG 到系统日志文件，然后再丢弃。

```
iptables -N LOG_DROP
iptables -A LOG_DROP -j LOG --log-tcp-options --log-ip-options --log-prefix '[IPTABLES DROP] : '
iptables -A LOG_DROP -j DROP
```

接着修改所有规则，只要是需进行 DROP 动作，都改为跳到 LOG_DROP 规则链，例如:

```
$IPTABLES -A bad_tcp_packets -p TCP ! --syn -m state --state NEW -j DROP
```

改为

```
$IPTABLES -A bad_tcp_packets -p TCP ! --syn -m state --state NEW -j LOG_DROP
```

至于需要进行 ACCEPT 处理的规则也如法炮制，先建立 LOG_ACCEPT 规则链:

```
iptables -N LOG_ACCEPT
iptables -A LOG_ACCEPT -j LOG --log-tcp-options --log-ip-options --log-prefix '[IPTABLES ACCEPT] : '
```

iptables -A LOG_ACCEPT -j ACCEPT
接着修改所有规则，例如:
$IPTABLES -A allowed -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
改为
$IPTABLES -A allowed -p TCP -m state --state ESTABLISHED,RELATED -j LOG_ACCEPT
安装显示分析结果的 PHP 程序:
在解压缩的数据夹中，找到一个叫做 web 的子数据夹，里面就是要给人从网页浏览分析结果的 PHP 程序，如果你的 Apache 已经设定好支持 php 和 php_mysql，那么只要将此数据夹复制到 Apache 的根文件目录就行了。方法如下:
cp -R web /var/www/iptables
拷贝完成后请修改 config.php ，目的是为了让 PHP 程序执行时，能以正确的账号密码连上 MySql ，以便从数据库读取数据，请找到底下三行:
$db_host="localhost";(一般不需修改，除非数据库在另一台主机上)
$db_user="iptables_user";(修改为仅具有读取权限的账号，如果之前安装数据库有自设帐号的话)
$db_password="xxxx";(请将密码修改为自设的密码)
安装 feed_db.pl:
前面已经介绍过这支程序的作用，请务必修改这支程序中有关数据库联机的 SQL 指令，将指令中的账号密码，改成你当初所设定的账号密码，建议最好不要用 root 身分联机，以免影响防火墙的安全性。这支程序是放在解压缩后数据夹内的 scripts 子数据夹，请修改下面这三行:
my $dsn = 'DBI:mysql:iptables:localhost';(请将 iptables 改成你自订的数据库名称)
my $db_user_name = 'iptables_admin';(请将 iptables_admin 改成你自订的管理账号)
my $db_password = 'xxxx';(请将 xxxx 改成管理账号的密码)
程序修改好后，请将它拷贝到 /usr/local/bin 数据夹，接着将程序执行起来，注意:这支程序会跑一个无穷尽循环，持续分析系统 LOG，因此必须在背景执行，同时只能有一支程序执行，以避免造成 IO 过大的负载，执行方式如下:
tail --follow=name --retry /var/log/syslog | /usr/local/bin/feed_db.pl &
如果这些动作都作了，但程序并未执行成功，有可能是因为 perl_DBI:DBD 套件没安装，请自行用 rpm 补装该套件。
查看分析结果:
完成以上所有安装步骤后，就可以坐下来享受一下成果，请打开浏览器输入底下网址
http://192.xx.1.254/iptables
看到的画面，如下图所示(请点选放大):

安装 LOG 分析后的安全防范:
防火墙上安装越多套件，系统安全性也就越低，改进的方法有两个:
1. 将 apache 和 mysql 架设在内部网络的一台机器上，防火墙上仅安装 mysql client，这样可以避免 apache 和 mysql 的漏洞被骇客利用
2. 在防火墙上设定规则，仅允许来自内部的网络，进行 HTTP 和 MySql 联机，这个方法比较简单，本文所介绍的 iptables 范例也是采用此法，缺点是无法在校外查看 LOG 分析结果

792 ☆NAT 禁止访问某些网站

iptables -I FORWARD -d www.playboy.com -j DROP

793 ☆NAT 禁止客户机上网

iptables -I FORWARD -s 192.168.152.12800 -j DROP

794 ☆NAT 禁止客户机访问某些服务

iptables -I FORWARD -s 192.168.0.0/24 -p tcp --dport 21 -j DROP

795 ☆NAT 强制范围某些站点

iptables -t nat PREROUTING -i eht0 -p tcp --dport 80 -j DNAT --to 210.21.34.33:80
这里只能用 ip,不能为虚拟主机的 ip

796 ☆NAT 禁止 ICMP

iptables -I INPUT -i ppp0 -p icmp -j DROP

797 ☆NAT 发布内部网络服务器

iptables -t nat PREROUTING -i ppp0 -p tcp --dport 80 -j DNAT --to-destination 192.168.152.128:80

798 ☆NAT 智能 DNS

iptables -t nat PREROUTING -i eth0 -p tcp --dport 53 -j DNAT --to-destination 192.168.152.128:53
iptables -t nat PREROUTING -i eth0 -p udp --dport 53 -j DNAT --to-destination 192.168.152.128:53

799 ☆PAM

Linux-PAM(Pluggable Authentication Modules).是由 Sun 提出的一种认证机制.它通过提供一些动态链接库和一套统一的 API,将系统提供的服务和该服务的认证方式分开,使得系统管理员可以灵活地根据需要给不同的服务配置不同的认证方式而无需更改服务程序,同时也便于向系统中添加新的认证手段.

pam 主要是由一组共享库文件（share libraries, 也就是.so 文件）和一些配置文件组成的。当你在请求服务的时候，具有 pam 认证功能的应用程序将与这些.so 文件进行交互，以便得知是否可以授权给发起请求的用户来使用服务.
如果认证成功了，那么这个用户便可以使用服务或完成命令，如果认证失败了，那么这个用户将不能使用服务，同时，pam 将向指定的 log 文件写入警告信息。我们可以将 pam 看作是一个中间

裁判，它不依赖于任何应用或服务。你完全可以升级这些应用或服务而不必管 pam 的共享库的更新或升级，反之亦然。所以它非常的灵活。
pam 的认证过程是通过对一些服务或应用的配置文件来控制的。通常，这些配置文件在/etc/pam.d 目录下。cd 到这个目录，你会看到很多你很熟悉的应用或服务的名称。

系统管理员通过 pam 配置文件来制定认证策略，即指定什么服务该采用什么样的认证方法;
应用程序开发者通过在服务程序中使用 pam API 而实现对认证方法的调用;
而 pam 服务模块（service module）的开发者则利用 pam SPI（Service Module API）来编写认证模块（主要是引出一些函数 pam_sm_xxxx( )供 libpam 调用），将不同的认证机制（比如传统的 UNIX 认证方法、Kerberos 等）加入到系统中;
pam 核心库（libpam）则读取配置文件，以此为根据将服务程序和相应的认证方法联系起来。

800 ☆PAM 的文件

/lib/libpam.so.*
pam 核心库

/etc/pam.d/或者/etc/pam.conf
pam 配置文件

/lib/security/pam_*.so
可动态加载的 pam service module

801 ☆PAM 配置

pam 的配置是通过单个配置文件/etc/pam.conf
RedHat 还支持另外一种配置方式，即通过配置目录/etc/pam.d/，且这种的优先级要高于单个配置文件的方式。
该目录下的每个文件的名字对应服务名
如果名为 xxxx 的服务所对应的配置文件/etc/pam.d/xxxx 不存在，则该服务将使用默认的配置文件/etc/pam.d/other

802 ☆PAM 配置文件的格式

Linux-pam 特有的符号是不区分大小写的. 而模块的路径,是大小写敏感的,因为它标识的是 Linux 下的文件的名字. 而任何模块参数的大小写分别由各个模块定义.
还有两个特殊的字符: 注解由'#'开头,结束于行结束,另外,模块的描述行可以以 '\'脱字符延续到下一行.

service module-type control_flag module_path args

service--由于现代的 pam 配置文件都是在 pam.d 下以服务程序的名字分开配置，所以基本上 service 这个项已经省去了。但在有些还使用一个 pam.conf 大文件配置所有的 pam 服务的 BSD 上，还有这一项。比如 telnet、login、ftp 等，服务名字“OTHER”代表所有没有在该文件中明确配置的其它服务。

module_type--控制使用什么类型的管理机制。同一个服务可以调用多个 pam 模块进行认证，这些模块构成一个 stack。
auth--认证管理（authentication management）
主要是接受用户名和密码，进而对该用户的密码进行认证，并负责设置用户的一些秘密信息。做两件事：1.检验用户是否是真正声称的那个用户,比如，要求用户输入用户名和密码来声明自己。2.这个模块可以通过 pam 自身的印证机制赋予用户一些特权，或是加入组。
account--帐户管理（account management）
主要是检查帐户是否被允许登录系统，帐号是否已经过期，帐号的登录是否有时间段的限制等等。是非认证式的对用户赋予／阻止使用一些系统资源。比如说，用户登录的时间限制，密码的期限等。
password--密码管理（password management）
主要是用来修改用户的密码。是用来管理用户的密码认证标记(token)的。比如，用户密码的尝试次数等。
session--会话管理（session management）
主要是提供对会话的管理和记账（accounting）。比如在会话前中后所要执行的一些事情，如记录会话信息,对如可展开会话的控制等。

control_flag--可以理解成对 pam 认证的流程控制。比如说如果成功，下一步应该怎么办？如果不成功，又应该怎么办？还有就是按怎样的顺序来进行认证，等等。
required--表示这一模块的认证是必须成功的，但如果失败，认证过程不会即刻终止，pam 将继续下一个同类型的模块认证。
requisite--和 required 类似，只是如果失败，认证过程将立即终止。
sufficient--表示如果认证成功，那么对这一类的模块认证是充足的了，其他的模块将不会再检验。
optional--表示这一模块认证是可选的，也不会对认证成功或失败产生影响。我没怎么用过这个。
include--有点类似于 DNS 或 xinetd 的 include。表示将包括其他的一些认证，这样就可以建立一个分离式的配置文件管理机制。

module_path--使用的认证模块的位置，最为保险的是使用绝对路径。不过 pam 有自己的 PATH 环境，在 RHEL4 中，大部分是使用的相对路径。如果你是缺省安装，有没有挪动过那些.so 文件的位置，使用相对路径是没问题的。(缺省路径一般是：/lib/security)

args--在认证时传给模块的一些变量。比如用什么文件，对于什么 uid,失败的处理等等。类似于赋给一个命令一些参数。
一般来说每个模块的参数都不相同，可以由该模块的开发者自己定义，但是也有以下几个共同的参数：
debug
该模块应当用 syslog( )将调试信息写入到系统日志文件中。

no_warn
表明该模块不应把警告 warning 信息发送给应用程序。
use_first_pass
表明该模块不能提示用户输入密码，而应使用前一个模块从用户那里得到的密码。(这模块只对 auth 和 password 模块有意义)
try_first_pass
表明该模块首先应当使用前一个模块从用户那里得到的密码，如果该密码验证不通过，再提示用户输入新的密码。(这模块只对 auth 和 password 模块有意义.)
use_mapped_pass
该模块不能提示用户输入密码，而是使用映射过的密码。即利用主密码将加密过的次密码解密出来并进行认证。(这模块只对 auth 和 password 模块有意义.)
try_mapped_pass
这个选项指示本模块首先尝试使用映射过的密码，即利用主密码将加密过的次密码解密出来并进行认证。如果密码认证失败，则再提示用户输入密码。(这模块只对 auth 和 password 模块有意义.)
expose_account
允许该模块显示用户的帐号名等信息，一般只能在安全的环境下使用，因为泄漏用户名会对安全造成一定程度的威胁。

其中的密码映射（password-mapping）允许用户在不同的认证机制下使用不同的密码，其中有一个主密码（primary password），其它密码为次密码（secondary passwords，可能有多个）。主密码用来对次密码进行加密。在主密码认证通过后，认证模块利用主密码将加密过的次密码（也称为 mapped password）解密，并对次密码进行认证。
注：如果使用了一次性密码的机制，就不使用密码映射。

还有一种比较复杂的语法来设置控制标志 control_flag，它由一组 value=action 形式的标记组成，标记之间以空格分开：
[value1=action1 value2=action2 ...]

这里的 valueI 有以下值 return values: success; open_err; symbol_err; service_err; system_err; buf_err; perm_denied; auth_err; cred_insufficient; authinfo_unavail; user_unknown; maxtries; new_authtok_reqd; acct_expired; session_err; cred_unavail; cred_expired; cred_err; no_module_data; conv_err; authtok_err; authtok_recover_err; authtok_lock_busy; authtok_disable_aging; try_again; ignore; abort; authtok_expired; module_unknown; bad_item; conv_again; incomplete; 和 default. 最后的 (default) 被用来定义当没有明确定义时的默认动作.

actionI--可以是一个正数或者是以下标识: ignore; ok; done; bad; die; 和 reset. 当以一个正数 I 作为 action 时,它的作用是指示以下 J 个模块将被跳过. 通过这种手段, 管理员可以开发出适度复杂的模块堆叠,它以许多不同的路径执行. 至于以何种路径则决定于某个模块的反应.

ignore--此类模块的返回状态将不会影响应用程序所得到的返回值.
bad--这表示相应的返回值将被认为是模块失败. 如果此模块是堆叠中的第一个失败的模块, 它的状态值将作为整个堆叠的状态.

die--和 bad 相同,不过会终止整个模块堆叠,pam 立即返回到应用程序.
ok--这告诉 pam 管理员要让此返回值直接作用于整个模块堆叠的返回. 换句话说, 如果堆叠的原先状态会导致返回 pam_SUCCESS, 这模块的返回值将会覆盖这结果. 请注意:如果堆叠的原先状态保存着一些意为模块失败的值,这'ok'将不会用来覆盖那样的值.
done--和 ok 一样, 不过会终止整个模块堆叠, pam 立即返回到应用程序.
reset--清除储存模块堆叠状态的内存并且重新开始下一组堆叠.

803 ☆PAM 认证模块

pam_access.so--缺省配置文件是/etc/security/access.conf （当然，通过“accessfile=/path/to/file” argument, 还可以自行指定配置文件。 通过加入这个认证模块到你想要控制的服务器 pam 配置文件，你可以实现对某些服务的 userbase 级控制。如 vsftp, samba,等。编辑/etc/security/access.conf 文件，加入你想要控制的用户，可以赋予/阻止他们从特定的来源登录服务器。
account required /lib/security/pam_access.so accessfile=/etc/security/access.conf

pam_cracklib.so--用字典方式检测 password 的安全性。有一些很有用的 arguments, 比如准许 retry 的次数，多少个字符可以和上次的密码重复，最小的密码长度等等。对密码的强度进行一定的检查库文件 libcrack 和字典文件/usr/lib/cracklib_dict

pam_deny.so--一个特殊的 pam 模块，这个模块将永远返回否。类似大多数的安全机制配置准则，一个严谨的安全规则的最后一项永远是否。

pam_limits.so--限制用户会话所能使用的系统资源 /etc/security/limits.conf.类似 Linux 的 ulimit 命令，赋给用户登录某个会话的资源限度。如 core 文件的大小，memory 的用量，process 的用量，等等。

pam_listfile.so--根据指定的某个文件决定是否允许或禁止提供服务,高度自定义的一个认证模块。理论上你可以用这个模块来控制任何服务。Arguments 有：
item=[tty|user|rhost|ruser|group|shell]
sense=[allow|deny]
file=/path/to/the/file
apply=[usre|@group]
例如:auth required pam_listfile.so item=user sense=deny file=/etc/security/dovecot.deny onerr=fail
然后我建立/etc/security/dovecot.deny 文件，每个用户占一行，在里面我加上 userA 的用户名。这样，userA 就无法登录 pop3 服务器了。原因是“sense＝deny “。当然，你也可以配置成“sense＝allow “，这样在这个文件里的用户就成了可以登录的用户了，不过你得加入所有你想赋予登录权限的用户，因为不在这个文件中的用户将被 DENY！
Item=user 就表明是通过用户名进行控制，sense= allow 表示如果用户名出现在 /etc/security/dovecot.deny 文件中就返回认证成功信息，file=/etc/security/dovecot.deny 指定配置文件，onerr=fail 表示如果出现某些错误(比如无法打开配置文件等)时返回的结果，这里是失败信息。

pam_permit.so--和 pam_deny.so 正好相反，这个模块将永远返回真。

pam_rootok.so--root 用户将通过认证。如果是 root 用户则无条件地通过认证。

pam_securetty.so--将用/etc/securetty 文件来检测 root 用户的登录来源，不在 securetty 文件中的来源一律禁止！

pam_tally.so--主要用来记录，重置，和阻止失败的登录（次数）。

pam_wheel.so--如果有这个模块，那个只有在 wheel 组里的用户可以得到 root 权限。

pam_xauth.so--如果有这个模块，那么在 su, sudo 的时候，xauth 的 cookies 将同时传到那个用户。（是不是想起来什么了？）

pam_pwdb--作为 pam_unix_xxxx 模块的一个替代。/etc/pwdb.conf 使用 Password Database 通用接口进行认证。

pam_radius--提供远程身份验证拨入用户服务(RADIUS)的认证

pam_rhosts_auth--利用文件~/.rhosts 和 /etc/hosts.equiv 和~/.rhosts /etc/hosts.equiv 对用户进行认证。

pam_securetty--提供标准的 Unix securetty 检查 /etc/securetty

pam_time--提供基于时间的控制，比如限制 /etc/security/time.conf 用户只能在某个时间段内才能登录

pam_unix--提供标准的 Unix 认证 /etc/passwd 和 /etc/shadow

pam_userdb--利用 Berkeley DB 数据库来检查 Berkeley DB 用户/密码

pam_warn--利用 syslog( )记录一条告警信息

pam_chroot--提供类似 chroot 命令的功能

pam_env--设置或取消环境变量 /etc/security/pam_env.conf

pam_filter--对输入输出流进行过滤 filters

pam_ftp.so--对匿名 ftp 用户进行认证

pam_group--当用户在指定的终端上请求指定的 /etc/security/group.conf 服务时赋予该用户相应的组权限

pam_issue--在提示用户输入用户名之前显示/etc/issue 文件的内容

pam_krb4--对用户密码进行 Kerberos 认证 相应的 Kerberos 库文件

pam_lastlog--在用户登录成功后显示关于 /var/log/lastlog 用户上次登录的信息，并维护 /var/log/lastlog 文件。

pam_mail--检查用户的邮箱中是否有新邮件 /var/spool/mail/xxxx

pam_mkhomedir--为用户建立主目录 /etc/skel/

pam_motd--显示/etc/motd 文件的内容 /etc/motd

pam_nologin--根据/etc/nologin 文件的存在与否 /etc/nologin 来决定用户认证是否成功

PAM 模块集合
6.1 access 模块
概要

模块名:

pam_access
作者:

Alexei Nogin <alexei@nogin.dnttm.ru>
维护者:

作者
提供的管理组:

account
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

需要一个设定档:/etc/security/access.conf
网络接口:

如果设定了就通过 pam_TTY , 否则尝试呼叫 ttyname()从标准输入文件描述符得到 tty 名字.标准的 gethostname(), yp_get_default_domain(), gethostbyname() 呼叫. NIS 用来提供网络用户组的支持.

概述

提供 logdaimon(?)类型的登录访问控制.
Account component

识别的参数:
描述:

这模块提供基于登录名称和主机(或域名),internet 地址(或网络数),或者非网络登录的终端名字的录访问控制.诊断信息通过 syslog(3)记录. 来自 Wietse Venema 的 logdaemon-5.6 里的 login_access.c 经过 A. Nogin 修改后采用.
例子/建议用法:

推荐使用. 例如,在管理像 NIS 服务器和邮件服务器时, 你需要有一些活动的帐号, 但却不想所有这些帐号都有登录的权限.

对于/etc/pam.d 类型的设定方式,你的模块放在/lib/security 里, 把下面一行加在 /etc/pam.d/login, /etc/pam.d/rlogin, /etc/pam.d/rsh 和 /etc/pam.d/ftp 前面:

account required /lib/security/pam_access.so

注意, 除非你的系统忽略.rhosts, 否则这模块不会产生作用. 见 pam_rhosts_auth 的文档.

在发布包里包含有一个 access.conf 的样本.

6.2 Chroot
概要

模块名:

pam_chroot
作者:

Bruce Campbell <brucec@humbug.org.au>
维护者:

Author; proposed on 20/11/96 - email for status
提供的管理组:

account; session; authentication
Cryptographically sensitive:

安全等级:
代码清洁度:

Unwritten.
系统依赖:
网络接口:

要求是 localhost.

概述

这模块的目的是对一般用户进行透明的包裹, 这可以把他们置于一个伪装的文件系统 (比如, 他们的'/' 实际上是 /some/where/else).

如果你有几个不同类的用户, 并且你对于安全性有一点点偏执, 那这模块将非常有用. 这可以用来限制哪些用户能看到什么, 并且限制他们只能执行哪些程序.
Account component:

Need more info here.
Authentication component:

Need more info here.
Session component:

Need more info here.

识别的参数:

Arguments and logging levels for the pam version are being worked on.
描述:
范例/建议用法:

提供一组合理的程序 - 仅放入 'cat', 'ls', 'rm', 'cp' 和 'ed' 有点...

不要太极端 (比如, 你可以为每个用户设定各自分开的环境, 但这样会太浪费磁盘空间.)

6.3 Cracklib 可插入的密码强度检查程序
概要

模块名:

pam_cracklib

作者:

Cristian Gafton <gafton@redhat.com>
维护者:

作者.
提供的管理组:

password
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

需要系统库 libcrack 和系统字典: /var/cache/cracklib/cracklib_dict.
网络接口:

概述

这模块可以插入给定程序的 password 堆叠来提供密码的强度检验.

这模块以以下流程运行: 它首先呼叫 Cracklib 例程来检查密码的强度; 如果 Cracklib 认可, 那么模块会经行额外一些检查. 这包括:

* Palindrome - 新密码是否为旧密码的回文?
* Case Change Only - 新密码是否只是拿旧密码改了些大小写?
* Similar - 新密码是否太像旧密码? 这由一个参数控制, difok 这是新旧密码的相差字符能被接受的最小个数, 模式是 10 个或者新密码的 1/2 长, 二者取小的那个.
* Simple - 新密码是否太简单? 这由五个参数控制: minlen, dcredit, ucredit, lcredit 和 ocredit. 这些参数如何生效以及默认值是什么见参数一节.
* Rotated - 新密码是否为旧密码的轮转?
* Already used - 密码是否曾经用过? 先前用过的密码保存在/etc/security/opasswd 里.

不带参数的配置这模块可以很好的工作在标准的 unix 密码加密环境. 对于 MD5 加密, 密码可以长于 8 个字符,那么这模块的默认设定就使用户选择一个满意的密码变得困难了. 显然, 对于新密码不能包含多于一半的旧密码的字符的要求就成了一个不一般的限制了. 比如, 旧密码"the quick brow fox jumped over the lazy dogs" 将难于更改... 此外, 默认设定允许密码短至 5 个字符. 对于采用 md5 加密算法的系统, 加大允许的最短密码长度会是个好主意. One can then allow more credit for different kinds of characters but accept that the new password may share most of these characters with the old password.
Password component

识别的参数:

debug; type=XXX; retry=N; difok=N; minlen=N; dcredit=N; ucredit=N; lcredit=N; ocredit=N;
描述:

这部件的动作是提示用户输入密码然后通过比对系统字典和一连串规则来检查它的强度以识别出差劲的选择.

默认动作是提示用户输入一个密码, 检查强度, 然后, 如果通过检查, 提示第二次输入密码 (用来检验第一次的输入是正确的). 这之后, 密码被当作新的认证标识传送给随后的安装的模块.

默认的动作可以用一些参数改成许多不同的方式:

* debug - 这选项令此模块写入信息到 syslog(3)以显示此模块的现为(此选项不会 把密码信息写入日志档).
* type=XXX - 提示用户输入密码的默认提示符是: ``New UNIX password: " 和 ``Retype UNIX password: ". 用此参数可以把 "UNIX" 替换成 XXX.
* retry=N - 本模组要求新密码(用来检查强度)的默认次数是 1 次. 用这参数可以增加到 N 次.
* difok=N - 这参数会改变允许新旧密码的字符差异为至少 10 个这样的默认值. 此外, 如果有一半的字符不同, 新密码就会被采用.
* minlen=N - 可接受的最短的密码长度+1. 此外对于新密码的长度, 对于每个不同种的字符 (其它,大写,小写 和数字),会得到"加分" (长度+1). 此参数的默认值是 9, 这对于旧的 UNIX 密码体系已经够了, 但是对于利用 md5 的额外安全性或许太小了. 注意, Cracklib 本身有一对长度的限制: 4, 写死在程序里的"路太短"的限制;6,定意的限制.这两个限制会不经过参考 minlen 而被检查出来. 如果你要允许密码长度短至 5, 你要么不要使用此模块要么重新编译 crack 库并重新编译此模块.
* dcredit=N - 这是对于新密码因包含数字而得到"加分"的限制. 如果你有小于或等于 N 个数字, 每个数字将会对整个长度+1 来迎合当前 minlen 的值. dcredit 的默认是 1, 对于小于 10 的 minlen 推荐此值.
* ucredit=N - 这是对于新密码因包含大写字符而得到"加分"的限制. 如果你有小于或等于 N 个大写字符, 每个大写字符将会对整个长度+1 来迎合当前 minlen 的值. ucredit 的默认是 1, 对于小于 10 的 minlen 推荐此值.
* lcredit=N - 这是对于新密码因包含小写字符而得到"加分"的限制. 如果你有小于或等于 N 个小写字符, 每个小写字符将会对整个长度+1 来迎合当前 minlen 的值. lcredit 的默认是 1, 对于小于 10 的 minlen 推荐此值.
* ocredit=N - 这是对于新密码因包含其它字符而得到"加分"的限制. 如果你有小于或等于 N 个其它字符, 每个其它字符将会对整个长度+1 来迎合当前 minlen 的值. ocredit 的默认是 1, 对于小于 10 的 minlen 推荐此值.

范例/建议用法:

为举例说明如何使用此模块, 我们列出如何将此模块和 pam_unix 部件堆叠:

#
下面两行堆叠了两个 password 类型的模块. 在这例子里, 用户有 3 次机会输入健壮的密码.
"use_authok"参数确保 pam_unix 模块不再另外提示输入密码,
而是采用由 pam_cracklib 提供的密码.
#
passwd password required pam_cracklib.so retry=3
passwd password required pam_unix.so use_authtok

另外一个例子(以/etc/pam.d/passwd 的格式)是用 md5 密码加密的情形:

#%pam-1.0
#
下面两行采用 MD5 系统, 密码长度最少 14+出现数字的最多 2 个"加分"+
出现其它字符的最多 2 个"加分", 并且至少有 3 个字符没有出现在旧密码中.
#
password required pam_cracklib.so \
difok=3 minlen=15 dcredit= 2 ocredit=2
password required pam_unix.so use_authtok nullok md5

6.4 死锁模块
概要

模块名:

pam_deny
作者:

Andrew G. Morgan <morgan@parc.power.net>
维护者:

current Linux-pam maintainer
提供的管理组:

account; authentication; password; session
Cryptographically sensitive:
安全等级:
代码清洁度:

清洁的.
系统依赖:
网络接口:

概述

这模块用来拒绝访问. 它永远透过 pam 架构告知程序失败. 正如概述 above 里提到的, 这模块适合用在默认(OTHER)条目.
Account component

识别的参数:
描述:

这部件除了返回失败外不做任何动作, 返回类型是 pam_ACCT_EXPIRED.
范例/建议用法:

把这模块堆叠在 account 里将会阻止用户通过应用程序(引用 Linux-pam 的帐户管理函数 pam_acct_mgmt())获取访问.

下面的例子使登入变得不可能:

```
#
# add this line to your other login entries to disable all accounts
#
login account required pam_deny.so
```

Authentication component

识别的参数:
描述:

这部件除了返回失败什么也不做, 当 pam_authenticate()被呼叫(当应用程式试图认证用户时)时返回值是 pam_AUTH_ERR; 当 pam_setcred()被呼叫(当建立连接并设置用户证书时 -- 在具体实现中这函数不太可能被呼叫)时返回 pam_CRED_UNAVAIL .
范本/建议用法:

为拒绝默认程序的访问, 在你的 Linux-pam 里包含下面一行:

```
#
# add this line to your existing OTHER entries to prevent
# authentication succeeding with default applications.
#
OTHER auth required pam_deny.so
```

Password component

识别的参数:
描述:

模块的这部件会阻止用户有机会修改密码. 它总是返回 pam_AUTHOK_ERR.
范本/建议用法:

这模块可以用来阻止应用程序更新申请者的密码. 比如, 为阻止 login 在用户的旧密码过期后自动提示输入新密码, 应该把下面一行放在你的设定档里:

```
#
# add this line to your other login entries to prevent the login
# application from being able to change the user's password.
#
login password required pam_deny.so
```

Session component

识别的参数:
描述:

模块的这一方面会阻止应用程序在主机上开启会话.
范本/建议用法:

和其它的 session 模块一起工作, 那模块也许显示一下"当日消息". 这模块可以用来阻止用户开启 shell. 如有 pam_motd 在先, 我们或可用下面的设置来阻止用户登录并提示用户现在是系统维护时间:

```
#
# An example to see how to configure login to refuse the user a
# session (politely)
#
login session required pam_motd.so \
file=/etc/system_time
login session required pam_deny.so
```

6.5 Set/unset 环境变量
概要

模块名:

pam_env
作者:

Dave Kinchlea <kinch@kinch.ark.com>
维护者:

Author
提供的管理组:

Authentication (setcred)
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

/etc/security/pam_env.conf
网络接口:

概述

这模块允许(取消)设定环境变量. Supported is the use of previously set environment variables as well as pam_ITEMs such as pam_RHOST.
Authentication component

识别的参数:

debug; conffile=configuration-file-name; envfile=/env-file-name; readenv=/0|1
描述:

这模块允许你(取消)设定任意的环境变量为固定字串, 先前的环境参数和/或 pam_ITEM.

通过设定档(默认是,/etc/security/pam_env.conf, 但是可由 connfile 参数改变) 进行所有的控制. 每行由变量名开始, 每个变量有两个可选项: DEFAULT 和 OVERRIDE. DEFAULT 用来设定变量的默认值, 如果没有指定默认值则设为空字串. OVERRIDE 告诉 pam_env 如果变量以设定就覆盖(覆盖" "默认值). 如果未设定 OVERRIDE, 则假定为""并且不会覆盖其它值.

VARIABLE [DEFAULT=[value]] [OVERRIDE=[value]]

(也许不存在的)环境变量可以指定值为${string} 并且(也许不存在的)pam_ITEM 可以用来指定值:&{string} . $和& 可以由反斜线脱意为一般字符 (成为\$本生). 双引号可以用来封装有空格的值 (但是不可以用作变量名)必须整个被引号包含并且里面不应该有引号或者脱意的引号.

这模块还可以解析包含单纯的每行是键=值格式的文件(默认是/etc/environment). 可以用 envfile 旗标设定默认文件并且可以设定 readenv 旗标为 1 或 0 来决定是否启用这选项.

这模块的行为可以由以下旗标来更改: flags:

* debug - 往 syslog(3)写更多的信息.
* conffile=filename - 默认的设定档是/etc/security/pam_env.conf. 这选项可以覆盖此默认. 必须提供完整的文件名,目录+文件名.
* envfile=filename - 默认从/etc/environment 里读键=值对来设定变量. 这选项覆盖默认文件. 必须提供完整的文件名,目录+文件名.
* readenv=0|1 - 打开或关闭读取 envfile 指定的文件(0=关闭,1=开启). 默认是开启.

范本/建议用法:

更多的用法见 pam_env.conf .

6.6 filter 模块
概要

模块名:

pam_filter
作者:

Andrew G. Morgan <morgan@parc.power.net>
维护者:

作者.
提供的管理组:

account; authentication; password; session
Cryptographically sensitive:

Not yet.
安全等级:
代码清洁度:

在 Linux 系统下可以干净的被编译.
系统依赖:

需要安装 filters .
网络接口:

概述

这模块用来提供给类似 ttysnoop(需要一些参考)的程序一个可选的插件. 因为还没有为这功能所写的插件出现, 这模块目前只是个玩具. 提供给这模块的唯一一个插件仅是对输入输出流做大小写转换. (这会变得非常讨厌并对基于 termcap 的编辑器不太友好)
Account+Authentication+Password+Session components

识别的参数:

debug; new_term; non_term; runX
描述:

每个组建都有呼叫过滤器的能力. 过滤器总是以所属应用程序的权限被 execv(2) 而 不是 以用户的权限. 因此这些过滤器并不总能被用户不经关掉相关会话而杀掉.

这模块的行为会被传给它的参数相当程度的左右:

* debug - 这选项增加模块被执行时写进 syslog(3)的信息量.
* new_term - 过滤器的默认动作是会设定 pam_TTY 为用户连上应用程序时所用的终端代号. 此参数指示过滤器可以设定 pam_TTY 为一个过滤了的伪终端.
* non_term - 不要试图设定 pam_TTY .
* runX - 模块必须得知道何时呼叫过滤器. 这参数告知过滤器何时被呼叫, 参数后面是各自的过滤器的完整路径和过滤器执行时的参数.

X 所允许的值是 1 和 2. 这指示过滤器执行的确切时间. 阅读 Linux-pam 的模块开发指南会对解释此观念有所帮助. 基本上, 针对每个管理组都有多至两个呼叫此模块的函数.

在 authentication 和 session 部件里, 实际上有两个不同的函数. 对于 authentication, 这两个函数是_authenticate 和_setcred -- 此时 run1 意味着在呼叫_authenticate 时执行过滤器而 run2 意味着呼叫_setcred 时运行过滤器. 对于 session 来说, run1 意指在_open_session 阶段运行过滤器, run2 在 _close_session 作用.

对于 account 部件. run1 和 run2 的任何一个被运行.

对于 password 部件, run1 用来指示当_chauthtok 第一次被执行(pam_PRELIM_CHECK 阶段)时呼叫过滤器,run2 指示第二次时(tt/pam_UPDATE_AUTHTOK/阶段)时呼叫.

范本/建议用法:

在写作本时, 这模块鲜有实际应用. 如兴趣所致, 你可以试着把下面几行加入你的 login 的设定.

#
An example to see how to configure login to transpose upper and

lower case letters once the user has logged in(!)
#
login session required pam_filter.so \
run1 /usr/sbin/pam_filter/upperLOWER

6.7 匿名访问模块
概要

模块名:

pam_ftp.so
作者:

Andrew G. Morgan <morgan@linux.kernel.org>
维护者:

作者.
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:
网络接口:

提示用户输入 email 地址; 易受欺骗的.(XXX - 需要加工)

概述

此模块的意图是提供一个可插入式的匿名 ftp 访问模式. mode of access.
Authentication component

识别的参数:

debug; users=XXX,YYY,...; ignore
描述:

此模块会拦截用户名和密码. 如果用户名是 ``ftp" 或 ``anonymous", 用户的密码以'@'为分隔符被分解成 pam_RUSER 和 pam_RHOST 两部分; these pam-items being set accordingly. The username is set to ``ftp". In this case the module succeeds. Alternatively, the module sets the pam_AUTHTOK item with the entered password and fails.

此模块的行为可以有以下旗标修改:

* debug - 往 syslog(3) 写更多的信息.
* users=XXX,YYY,... - 替代 ``ftp'' 或 ``anonymous'', 提供给以逗号分开的用户以匿名访问; ``XXX,YYY,...''. 申请人可以输入这些用户名之一, 返回的用户名设定为列表中的第一个用户;``XXX''.
* ignore - 不关心用户的 email 地址(如果提供).

范本/建议用法:

见 above.

6.8 群组访问模块
概要

模块名:

pam_group
作者:

Andrew G. Morgan <morgan@parc.power.net>
维护者:

作者.
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:

对于针对 setgid 状态的文件可访问性敏感.
代码清洁度:
系统依赖:

需要/etc/security/group.conf 文件. 可被编译成带有或不带 libpwdb.
网络接口:

仅通过正确的 pam_TTY 项.

概述

此模块提供基于用户名以及他们从哪个终端请求服务的群组设定. It takes note of the time of day.
Authentication component

识别的参数:
描述:

此模块不用于认证用户, 而是用来赋予用户组的身份(在认证模块的凭证设定阶段). 这些组身份基于他们请求的服务. 组身份列在/etc/security/group.conf .
范本/建议用法:

为使这模块正常运作,必需先要有格式正确的/etc/security/groups.conf . 此文件的格式如下.组身份基于服务应用程序满足此设定档的任意行来赋予. 没一行有如下格式(注解以`#'开头):

services ; ttys ; users ; times ; groups

前四个栏位和 pam_time 的 etc/security/pam_time.conf 语法相同, 最后的栏位, groups, 是以逗号(或者空格)分开的一个组的列表. 如果用户的应用程序满足前四个栏位, 用户就被赋予列表中的组身份.

通常, 这模块对于分配用户特有的文件访问权限有帮助. 问题是一旦用户得到了组身份, 他就可以创建一个 setgid 的属于某群组的程序. 这之后, 当用户不再是这个组的成员时, 他们可以通过这程序获取组身份. 之所以用户访问的文件系统如此重要, 是一旦文件系统被加载为 nosuid 时, 这样的程序就无法执行. 为使这模块提供任何级别的安全性, 用户能有写权限的所有文件系统都应该被加载成 nosuid 模式.

pam_group 模块的功能和/etc/group 平行. 如果用户已被此模块赋予了任何组身份, 他还另外被赋予/etc/group 里的相关组身份.

6.9 Add issue file to user prompt
概要

模块名:

pam_issue
作者:

Ben Collins <bcollins@debian.org>
维护者:

Author
提供的管理组:

Authentication (pam_sm_authenticate)
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:
网络接口:

概述

此模块在当提示输入用户名时显示发布文件(/etc/issue).
Authentication component

识别的参数:

issue=issue-file-name; noesc;
描述:

此模块允许你在提示用户输入用户名之前显示出发布信息. 默认状态下也会解析发布文件中的脱意代码,类似有些通用的 getty(\x 格式).

识别的脱意代码:

* d - 当前日期
* s - OS 名称
* l - 当前 tty 的名字
* m - 本系统的架构(i686, sparc, powerpc, ...)
* n - 主机名
* o - 域名
* r - OS 的版本发行号(例如 2.2.12)
* t - 当前时间
* u - 当前登入的用户数
* U - 类似 u, 它能区分用"user" 还是 "users" (比如. "1 user" or "10 users")
* v - OS 版本/构建日期(比如. "#3 Mon Aug 23 14:38:16 EDT 1999" on Linux).

以下旗标可以用来改变此模块的行为:

* issue - 替换默认发布文件
* noesc - 不解析脱意代码

范本/建议用法:

login auth pam_issue.so issue=/etc/issue

6.10 The Kerberos 4 module.
概要

模块名:

pam_krb4
作者:

Derrick J. Brashear <shadow@dementia.org>
维护者:

作者.
提供的管理组:

authentication; password; session
Cryptographically sensitive:

采用相关 API
安全等级:
代码清洁度:
系统依赖:

相关库- libkrb, libdes, libcom_err, libkadm; 和一组 Kerberos 的头文件.
网络接口:

从网络上的 Kerberos 的票据中心得到 Kerberos 的票据授权票.

概述

此模块提供了进行 Kerberos 密码认证的界面, 它能从 Kerberos 的票据授权服务器取得票据授权票, 离线时毁掉票据, 和修改 Kerberos 密码.
Session component

识别的参数:
描述:

此部件目前会设定环境变量 KRBTKFILE (虽然目前还没法 export 这变量), 以及当登出时删除用户的票据(要等到 login 支持 pam_CRED_DELETE).
范本/建议用法:

直到我们能通过 Linux-pam 改变环境.

Password component

识别的参数:

use_first_pass; try_first_pass
描述:

这部件改变用户的 Kerberos 密码, 它首先凭旧密码从密码修改服务得到活动钥匙, 然后发送新密码给那服务.
范本/建议用法:

仅能用于真实的 real Kerberos v4 kadmind. It cannot be used with an AFS kaserver unless special provisions are made. Contact the module author for more information.

Authentication component

识别的参数:

use_first_pass; try_first_pass
描述:

此模块通过从 Kerberos 服务器申请票据授权票来验证用户的 Kerberos 密码, 并且可选的如果本机键文件存在就从尝试此票据中获取本机的主机键并和本机的键文件比对.

它还会把票据记录在文件以便后序使用, 并且在当离线是删除票据文件(目前还不行,除非 login 会呼叫 pam_CRED_DELETE).
范本/建议用法:

此模块可以同采用 MIT v4 的 Kerberos 服务器协同工作. 可以加以修改已使其支持 AFS 类型的 Kerberos. 为防止密码算法的不一致,目前还没有支持.

6.11 The last login module
概要

模块名:

pam_lastlog
作者:

Andrew G. Morgan <morgan@parc.power.net>
维护者:

作者.
提供的管理组:

auth
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

用到/var/log/lastlog 里保存的信息.
网络接口:

概述

此模块用于维护/var/log/lastlog 文件. 当程序呼叫 pam_open_session()函数时写入一个条目并且在呼叫 pam_close_session()时完成此条目. 这函数还可以显示一条关于最后一次登录的信息. 如果应用程序已经有此功能, 则没有必要使用此模块.
Authentication component

识别的参数:

debug; nodate; noterm; nohost; silent; never
描述:

此模块用于无论用户从何种支援 pam 的应用程序登录系统都提供"Last login on ..." 这样的信息. 此外, 此模块还维护/var/log/lastlog 档.

可有一下旗标改变此模块的行为:

* debug - 往 syslog(3)里写入更多的信息.
* nodate - 当显示最后登录的信息时隐藏具体日期.
* noterm - 当显示最后登录的信息时隐藏终端名称.
* nohost - 当显示最后登录的信息时隐藏客户机名称.
* silent - 不显示任何最后登录信息: 仅仅更新/var/log/lastlog .
* never - 如果/var/log/lastlog 没有包含此用户的任何曾经登录的信息, 则显示用户从未登录的信息: ``welcome..." .

范本/建议用法:

此模块可以用来在用户 login 后显示是否有新邮件. 下面是一个以/etc/pam.conf 配置的样本:

#
do we have any mail?
#
login session optional pam_lastlog.so

注意, 有些应用程序也许自己会实现这些功能. 这种情况下就不必使用此模块了.

6.12 资源限制模块
概要

模块名:

pam_limits
Authors:

Cristian Gafton <gafton@redhat.com>
Thanks are also due to Elliot Lee <sopwith@redhat.com> for his comments on improving this module.
维护者:

Cristian Gafton - 1996/11/20
提供的管理组:

session
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

需要有/etc/security/limits.conf 设定档并且内核要支持资源限制. 也使用 libpwdb 库.
网络接口:

概述

此模块通过 Linux-pam 的 开启-会话的钩子来设定用户会话能够获取的资源的限制. 由下面会讨论的设定档来更精确的支配动作.
Session component

识别的参数:

debug; conf=/path/to/file.conf
描述:

通过设定设定档/etc/security/limits.conf 的内容来设定用户会话的资源限制. uid=0 的用户不受此约束.

以下旗标可以用来改变此模块的行为:

* debug - 往 syslog(3)写入冗长的记录.
* conf=/path/to/file.conf - 指定一个替换的 limits 设定档.

范本/建议用法:

为使用此模块, 系统管理员必须首先建立一个 root 只读 的文件(默认是 /etc/security/limits.conf). 这文件描述了 superuser 想强迫用户和用户组的资源限制. uid=0 的帐号不会受限制.

设定档的每一行描述了一个用户的限制,以下面的格式:

<domain> <type> <item> <value>

上面列出的栏位可以填下面的值:...
<domain> 可以是:

* 一个用户名
* 一个组名,语法是@group
* 通配符*, 定义默认条目

<type> 可以有一下两个值:

* hard 为施行硬 资源限制. 这些限制由 superuser 设定,由 Linux 内核施行. 用户不能提升他对资源的需求到大于此值.
* soft 为施行软 资源限制. 用户的限制能在软硬限制之间上下浮动. 这种限制在普通用法下可以看成是默认值.

<item> 可以是以下之一:

* core - 限制 core 文件的大小(KB)
* data - 最大的资料大小 (KB)
* fsize - 最大的文件尺寸 (KB)
* memlock - 最大能锁定的内存空间(KB)
* nofile - 最多能打开的文件
* rss - 最大的驻留程序大小(KB)
* stack - 最大的堆栈尺寸(KB)
* cpu - 最大的 CPU 时间(分钟)
* nproc - 最多的进程数

* as - 地址空间的限制
* maxlogins - 用户的最多登录数
* priority - 用户进程执行时的优先级

要完全不限制用户(或组), 可以用一个(-)(例如: ``bin -", ``@admin -"). 注意,个体的限制比组限制的优先级高, 所以如果你设定admin组不受限制, 但是组中的某个成员被设定档中某行限制, 那么此用户就会依据这样被限制.

还应该注意, 所有的限制设定只是每个登录的设定. 他们际不是全局的,也不是永久的 ; 之存在于会话期间.

在 limits 的设定档中, ``#" 开头的行为注解.

pam_limits 模块会通过 syslog(3)报告它从设定档中找到的问题.

下面是个设定档的样本:

```
# EXAMPLE /etc/security/limits.conf file:
# =======================================
# <domain> <type> <item> <value>
* soft core 0
* hard rss 10000
@student hard nproc 20
@faculty soft nproc 20
@faculty hard nproc 50
ftp hard nproc 0
@student - maxlogins 4
```

注意, 对同一个资源(见@faculty)的软限制和硬限制 - 这建立了用户可以从指定服务会话中得到的默认和最大允许的资源数.

对于需要进行资源限制的服务例程(如 login), 把下面一行放在/etc/pam.conf 里此服务的设定的最后一行(通常在 pam_unix session 后面):

```
#
# Resource limits imposed on login sessions via pam_limits
#
login session required pam_limits.so
```

6.13 The list-file module
概要

模块名:

pam_listfile
作者:

Elliot Lee <sopwith@cuc.edu>
维护者:

Red Hat Software:
Michael K. Johnson <johnsonm@redhat.com> 1996/11/18
(if unavailable, contact Elliot Lee <sopwith@cuc.edu>).
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:

clean
系统依赖:
网络接口:

概述

list-file 模块提供基于任意文件的对服务例程的拒绝或允许的设定
Authentication component

识别的参数:

onerr=succeed|fail; sense=allow|deny; file=filename; item=user|tty|rhost|ruser|group|shell
apply=user|@group
描述:

这模块先获取指定类型的项目 -- user 指定用户名, pam_USER; tty 指定请求建立时的获取的终端名, pam_TTY; rhost 指定请求建立的主机 (如果有), pam_RHOST; ruser 指定对方主机中发起连接请求的用户名(如果可以), pam_RUSER -- 然后寻找这些项目是否出现在 filename 中. filename 每行包含一个项目. 如果找到对应项目, 那么如果 sense=allow, 则返回 pam_SUCCESS, 认证请求成功; 否则如果 sense=deny, 就返回 pam_AUTH_ERR, 认证请求失败.

如果有错误发生(比方, 如果 filename 不存在, 或者参数不完整), 那么如果 onerr=succeed, 就返回 pam_SUCCESS , 否则如果 onerr=fail, 就返回 pam_AUTH_ERR 或者 pam_SERVICE_ERR.

附加的参数, apply=, 能够限制应用程序只对指定的用户 (apply=username)或指定的组 (apply=@groupname)适用本规则. 这个附加的限制只对 tty,rhost 和 shell 有意义.

除了这最后一个参数外, all arguments should be specified; do not count on any default behavior, as it is subject to change.

此模块不授予任何证书.
范本/建议用法:

传统的``ftpusers" 认证可以由/etc/pam.conf 下面的条目实现:

```
#
# deny ftp-access to users listed in the /etc/ftpusers file
#
ftp auth required pam_listfile.so \
onerr=succeed item=user sense=deny file=/etc/ftpusers
```

注意,列在/etc/ftpusers 里的用户将不允许访问 ftp 服务.

要只允许指定的用户可以登录, 你可以在 pam.conf 里设定以下的条目:

```
#
# permit login to users listed in /etc/loginusers
#
login auth required pam_listfile.so \
onerr=fail item=user sense=allow file=/etc/loginusers
```

要什这正常工作, 所有允许登录的用户都应该列在/etc/loginusers. 除非你明确的要试着把 root 所在门外, 否则在你这样设定好, 登出之前最好确认: 要么 root 列在/etc/loginusers, 要么列表里有用户可以 su 成 root.

6.14 The mail module
概要

模块名:

pam_mail
作者:

Andrew G. Morgan <morgan@linux.kernel.org>
维护者:

Author
提供的管理组:

Authentication (credential) Session (open)
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

默认的邮件目录:/var/spool/mail/
网络接口:

概述

此模块查看用户的邮件目录并指示里面是否有邮件. whether the user has any mail in it.
Session component

识别的参数:

debug; dir=direcory-name; nopen; close; noenv; empty; hash=hashcount; standard; quiet;
描述:

此模块给用户提供"you have new mail" 的服务. 它可以嵌入到任何有凭证钩子的应用程序里. 它给一个信息指示用户的邮件目录中是否有最新的邮件. 此模块还设定 Linux-pam 的环境变量 MAIL 为用户的邮件目录.

以下旗标可以用来改变此模块的行为:

* debug - 往 syslog(3)里写入更多信息.
* dir=pathname - 从给定的 pathname 查找用户的邮件. 默认的邮件位置是/var/spool/mail. 注意,如果提供的 pathname 以`~'开头, 意指目录是在用户的家目录里.
* nopen - 指示当用户获取证书时不要打印任何信息, 这旗标用来设定 MAIL 环境变量而不用显示任何信息.
* close - 指示模块当用户被撤回证书时检查是否有邮件.
* noenv - 不要设定 MAIL 环境变量.
* empty - 当用户的邮箱是空的时候显示这状态.
* hash=hashcount - 邮件目录的 hash 深度. 比如, hashcount = 2 意味着邮箱是 /var/spool/mail/u/s/user.
* standard - 旧的 "You have..." 格式, 不显示具体的邮箱位置. 同样实现"empty".
* quiet - 只在有新邮件时才报告.

范本/建议用法:

此模块用来当用户登录到系统是提示他有新邮件. 下面是/etc/pam.conf 的样本:

#
do we have any mail?
#
login session optional pam_mail.so

注意,有些应用程序会自己实现这功能, 这种情形下,就不必使用此模块了.

Authentication compent

authentication 部件和 session 部件一样运作, 除了他在 pam_setcred()阶段被激活.
6.15 在首次登录是创建用户家目录
概要

模块名:

pam_mkhomedir
作者:

Jason Gunthorpe <jgg@ualberta.ca>
维护者:

Ben Collins <bcollins@debian.org>
提供的管理组:

Session
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:
网络接口:

概述

为通过认证的用户按需创建家目录.
Session component

识别的参数:

debug; skel=skeleton-dir; umask=octal-umask;

描述:

此模块对于帐号集中管理(NIS,NIS+,或 LDAP)并且多系统访问的分布式系统有用处. 它不需要管理员在每个系统都去建用户的家目录,而是当用户第一次成功登录时自动创建. skeleton 目录(通常是/etc/skel/)用来拷贝默认文件, 这模块也为创建的文件设定 umask.

以下旗标可以用来改变此模块的行为:

* skel - 从 skeleton 目录处拷贝默认文件到新创建的家目录.
* umask - 和你在 shell 下设 umask 同样格式的八进制数.

范本/建议用法:

session required pam_mkhomedir.so skel=/etc/skel/ umask=0022

6.16 输出 motd 文件
概要

模块名:

pam_motd
作者:

Ben Collins <bcollins@debian.org>
维护者:

Author
提供的管理组:

Session (open)
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:
网络接口:

概述

此模块在成功登录时输出 motd 文件 (默认是/etc/motd) .
Session component

识别的参数:

debug; motd=motd-file-name;
描述:

此模块允许你当成功登录后有任意的 motd(当日消息)输出. 默认的文件是 /etc/motd, 但是可以设定为任意文件.

以下旗标可以用来改变此模块的行为:

* motd - 如果不打算用默认文件, 这里指定文件.

范本/建议用法:

login session pam_motd.so motd=/etc/motd

6.17 The no-login module
概要

模块名:

pam_nologin
作者:

Written by Michael K. Johnson <johnsonm@redhat.com>
(based on code taken from a module written by Andrew G. Morgan <morgan@parc.power.net>).
维护者:

Michael K. Johnson <johnsonm@redhat.com>
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:

1 个关于 dropping const 的警告
系统依赖:
网络接口:

概述

提供标准的 Unix nologin 的认证.

Authentication component

识别的参数:
描述:

提供标准的Unix nologin的认证. 如果存在/etc/nologin 文件, 则只有root允许登录; 其他用户会得到一个错误信息并且不能登入. 所有用户(root 或 其它用户) 会显示/etc/nologin里的内容.

如果不存在/etc/nologin , 此模块返回成功.
范本/建议用法:

为使此模块生效, 所有登录方法都要由此模块来把关. 为和传统UNIX的nologin相同作用, 它应该被作为required方法, 并且放在任何sufficient之前.

6.18 The promiscuous module
概要

模块名:

pam_permit
作者:

Andrew G. Morgan, <morgan@parc.power.net>
维护者:

Linux-pam 维护者.
提供的管理组:

account; authentication; password; session
Cryptographically sensitive:
安全等级:

非常低. 极度小心的使用.
代码清洁度:

Clean.
系统依赖:
网络接口:

概述

此模块非常不安全. 必须谨慎使用. 它的动作是永远允许访问. 不做任何其它事情.

Account+Authentication+Password+Session components

识别的参数:
描述:

不管用在哪种管理组, 这模块的动作只是简单的返回 pam_SUCCESS -- 操作成功.

在认证部件里, 用户名会被获取. 很多应用程序在不知道用户名时会觉得不合理.
范本/建议用法:

很难找到应用此模块的正当理由. 可是, 它还是有些合理的应用. 比如, 系统管理员想要关闭工作站的帐号管理, 同时又要允许登录, 那么她可以用下面的配置条目设定 ligin:

```
#
# add this line to your other login entries to disable account
# management, but continue to permit users to log in...
#
login account required pam_permit.so
```

6.19 The Password-Database module
概要

模块名:

pam_pwdb
作者:

Cristian Gafton <gafton@redhat.com>
and Andrew G. Morgan <morgan@linux.kernel.org>
维护者:

作者.
提供的管理组:

account; authentication; password; session
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

需要正确的配置 libpwdb
网络接口:

概述

此模块是 pam_unix_..的可插入式的替代品. 它使用 Password Database 的通用接口.
http://linux.kernel.org/morgan/libpwdb/index.html.
Account component

识别的参数:

debug
描述:

参数 debug 令此模块的帐号管理功能在动作时 syslog(3)更多的信息. (由别的功能支持的参数被忽略掉, 但是会在 syslog(3) 记录下错误信息.

基于以下的 pwdb_元件: expire; last_change; max_change; defer_change; warn_change, 此模块执行确认帐号和密码状态的任务. 至于后来, 它可以给用户提出修改密码的建议或者, 通过 pam_AUTHTOKEN_REQD 的返回, 延迟为用户提供服务直到他创建了一个新密码. 以上列出的条目在 Password Database Library Guide 里有解释. documented in the Password Database Library Guide (见上面的链接). 如果用户记录不包含一个或者更多这些条目, 对应的 shadow 检查不被执行.
范本/建议用法:

在帐号管理模式, 这模块可以象下面的用法:

```
#
# 确保用户帐号和密码依然有效
#
login account required pam_pwdb.so
```

Authentication component

识别的参数:

debug; use_first_pass; try_first_pass; nullok; nodelay; likeauth
描述:

参数 debug 令模块的认证功能写入 syslog(3)更多的信息.

如果用户的 official 密码为空, 此模块的默认动作是不允许用户访问服务. 参数 nullok 用来使此默认失效.

当给出参数 try_first_pass, 在提示用户输入密码之前, 模块会拿通过之前堆叠的 auth 模块认证的密码来试. 参数 use_first_pass 迫使模块使用上述的密码并且永远不会提示用户 - 如果没有可用的密码或密码不能通过认证, 用户将被禁止访问.

参数 nodelay 用来阻止认证部件在认证失败时的延时请求. 默认动作是请求一个一秒钟以上的失败-延迟.

由别的功能支持的参数被忽略掉, 但是会在 syslog(3) 记录下错误信息.

有个帮手程序, pwdb_chkpwd, 用来检查用户的密码(当密码报存在一个只读数据库时). 此程序非常简单,只检查当前用户的密码. 它被此模块的认证部件以当前用户的身份显式的呼叫. 这样象 xlock 这样的程序就不需被设成 setuid-root 而可以工作.

The likeauth argument makes the module return the same value when called as a credential setting module and an authentication module. This will help libpam take a sane path through the auth component of your configuration file.
范本/建议用法:

此模块的完整功能由一个恰当的/etc/pwdb.conf 来控制, 指定的用户数据库控制用户认证的来源.

Password component

识别的参数:

debug; nullok; not_set_pass; use_authtok; try_first_pass; use_first_pass; md5; bigcrypt; shadow; radius; unix
描述:

pam_pwdb 模块的这一部分执行更新用户密码的任务. 受惠于 libpwdb 的灵活性, 此模块可以把用户密码从一个数据库搬到另一个去, 也许以动态的方式使用户密码记录安全 (此时还处于初期 ALPHA 阶段!) - 这正是 shadow, radius 和 unix 参数的意图.

在传统的 UNIX 密码数据库(这里保存着用户的加密密码)的情况下, 参数 md5 用来采用 MD5 的加密算法以取代传统的 crypt(3)呼叫. 还有种选择, 参数 bigcrypt 采用 DEC(数字设备公司)的对标准 UNIX 的 crypt()的算法进行`C2'扩展的算法以加密超过 8 位长的密码.

参数 nullok 用来允许改变一个原先为空的密码. 如果没有此参数, 空的密码会当成是被锁住的帐号.

参数 use_first_pass 用来阻止选择旧的和新的密码,而是采用 password 里此模块之前的模块所保留的密码. 参数 try_first_pass 用来防止当 pam_pwdb 跟随另一个 password 模块时用户重复输入也许是共享的旧密码 - 如果旧密码不正确, 再提示用户输入. 参数 use_authok 用来迫使 此模块设定

新密码时采用 password 堆叠里先前的模块提供的值. (这用法在先前介绍的 Cracklib 一节里有个范本).

参数 not_set_pass 用来告诉模块不去关心给/从其它(堆叠的)密码模块提供/获取可用的旧密码或新密码.

参数 debug 使此模块的 password 功能 syslog(3)更多的动作的信息. 其它参数会往 syslog(3) 写进错误信息.
范本/建议用法:

有关如何将此模块与可插入式密码检查模块,pam_cracklib 一起堆叠的例子, 请见上面的 pam_cracklib 一节.

Session component

识别的参数:
描述:

模块的此部件没有参数. 它的作用就是在用户会话开始和结束时往 syslog(3) 里记录用户名和服务类别.
范本/建议用法:

session 部件的用法是单纯的:

```
#
# pwdb - unix like session opening and closing
#
login session required pam_pwdb.so
```

6.20 The RADIUS session module
概要

模块名:

pam_radius
作者:

Cristian Gafton <gafton@redhat.com>
维护者:

作者.
提供的管理组:

session
Cryptographically sensitive:

此模块不用来处理密码
安全等级:
代码清洁度:

在编译 /usr/include/rpc/clnt.h 时 gcc 报告一个警告. Hey, is not my fault !
系统依赖:
网络接口:

是的; 这是个网络模块(独立于应用程序).

概述

此模块用来提供用户从 RADIUS 服务器认证的会话服务. 目前阶段, 唯一提供的功能是以 RADIUS 服务器当一个帐号服务器来使用.
Session component

识别的参数:

debug - 冗余的信息写进 syslog(3).
描述:

此模块用来提供用户从 RADIUS 服务器认证的会话服务. 目前阶段, 唯一提供的功能是以 RADIUS 服务器当一个帐号服务器来使用.

(There are few things which needs to be cleared out first in the pam project until one will be able to use this module and expect it to magically start pppd in response to a RADIUS server command to use PPP for this user, or to initiate a telnet connection to another host, or to hang and call back the user using parameters provided in the RADIUS server response. Most of these things are better suited for the radius login application. I hope to make available Real Soon (tm) patches for the login apps to make it work this way.)

当开启一个会话时, 此模块发送一个``Accounting-Start" 消息给 RADIUS 服务器, 服务器将会记录/更新此用户的有关记录. 而当会话关闭时, 一个``Accounting-Stop" 消息会发给 RADIUS 服务器.

此模块无须其它的必要条件以使其工作. 兴趣所致, 你要安装一个 RADIUS 服务器, 然后以此服务器为一个集中的帐号管理服务器, 忘掉 wtmp/last/sac 这些东西.
范本/建议用法:

为使服务(例如 login)使用此模块, 把下面一行加入/etc/pam.conf, 放在此服务的最后一行(通常在 session 段的 pam_unix 之后):

login session required pam_radius.so

把 login 替换成你想要的其它服务名.

此模块大量使用了 libpwdb0.54preB 或其后版本的 API. 默认情况下, 它会从 /etc/raddb/server 读取 RADIUS 服务器的设定(主机名和加密方法). 这是 libpwdb 默认编译时的设定, 目前没有办法不经重编译 libpwdb 来改变这个默认设定. 我正在为 libpwdb 的 radius 支持进行扩展以使其提供运行时可设定功能.

还请注意 libpwdb 还需要你提供 RADIUS 的目录(/etc/raddb/dictionary).

6.21 The rhosts module
概要

模块名:

pam_rhosts_auth
作者:

Al Longyear <longyear@netcom.com>
维护者:
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:

干净.
系统依赖:
网络接口:

呼叫标准的 inet_addr(), gethostbyname() 函数.

概述

此模块为传统的如 rlogin 和 rsh 的服务提供基于网络的认证.
Authentication component

识别的参数:

no_hosts_equiv; no_rhosts; debug; no_warn; privategroup; promiscuous; suppress
描述:

此模块的认证机制基于两个文件: /etc/hosts.equiv (或为_PATH_HEQUIV 定意在#include <netdb.h>) 和 ~/.rhosts. 首先, 列在前一个文件里的主机会被当成 localhost 一般对待. 第二, 后一个用户自己的文件的条目用来对照"远程主机 远程用户"到当前主机的帐号. 如果远程主机列在 /etc/hosts.equiv,并且远程用户的帐号和本机用户一致或者远程用户的帐号在个人设定档中有登记则访问被允许.

用户的私人设定档会有些限制:它必须是一个常规文件(POSIX.1 定意为 S_ISREG(x)) ; 它必须属于 superuser 或本用户;出了所有者之外其它用户必须不可写.

此模块认证从远程主机(内部用 pam_RHOST 指定)连线的远端用户 (内部用 pam_RUSER 指定). 因此, 应用程序如果要兼容此认证模块, 它们必须在呼叫 pam_authenticate()之前设定好这些条目. 模块本生没有能力去独立的侦测这些网络连接信息.

至于 root 的访问, 除非使用 hosts_equiv_rootok 选项, /etc/host.equiv 将被忽略. 替代的是, superuser 必须有设定正确的私人设定档.

此模块的动作可有一下旗标修改:

* debug - 记录冗余的信息到 syslog(3). (XXX - 事实上, 此模块目前不记录任何信息, 请自愿的修复它!)
* no_warn - 不要给用户发送更多的有关失败的警告信息. (XXX - 此模块当前不会发布任何警告信息, 请自愿的修复它!)
* no_hosts_equiv - 忽略/etc/hosts.equiv .
* hosts_equiv_rootok - 允许 superuser 也使用 /etc/hosts.equiv. 如果没有这选项,/etc/hosts.equiv 不被 superuser 帐号考虑. 如果同时有 no_hosts_equiv, 此选项会失效.
* no_rhosts - 忽略所有用户的私人设定档:~/.rhosts.
* privategroup - 通常, ~/.rhosts 必须不能为除了所有者之外的任何人可写. 此选项针对组所有者和进行认证的拥有者同名的文件放宽组写的权限. 为减小此选项可能带来的安全隐患, 此模块还检查用户是否为那个组的唯一成员.
* promiscuous - 有此选项, 以`+' 作为主机名将导致所有的主机都被赋予访问权限. 没有此选项, `+'的记录会被忽略, debug 选项会往 syslog 里记录一个警告.
* suppress - 这将阻止模块在当认证失败时 syslog(3) 警告信息. 这选项主要用来使日志免于无意义的错误记录, 尤其是当模块被当成 sufficient 模式所用时.

范本/建议用法:

为允许用户从信任的远程主机登录, 你得把下面一行加入你的/etc/pam.conf, 并且在其他会提示用户输入密码的模块之前 :

```
#
# No passwords required for users from hosts listed above.
#
login auth sufficient pam_rhosts_auth.so no_rhosts
```

注意, 在这个例子里, 系统管理员关闭了所有个人的 rhosts 设定档. 还请注意, 此模块可用来设定成只允许在/etc/host.equiv 文件里指定的远程主机登录, 只要把 sufficient 改成 required.

6.22 The root access module
概要

模块名:

pam_rootok
作者:

Andrew G. Morgan <morgan@parc.power.net>
维护者:

Linux-pam maintainer
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:

Clean.
系统依赖:
网络接口:

概述

此模块用在 superuser 希望不需要输入密码就能获取访问权限的情形.
Authentication component

识别的参数:

debug

描述:

此模块认证 uid 是 0 的用户. 创建成 setuid-root 的应用程序通常保留用户的 uid 但是以有效的-uid 的权限执行. 真实的 uid 会被检查.
范本/建

【发表回复】【查看论坛原帖】【添加到收藏夹】【关闭】

---
ipod 回复于：2004-08-19 15:14:34
完全版？谢谢楼主先

---
gentoo 回复于：2004-08-19 15:46:27
谢谢!

---
thomassun 回复于：2004-08-20 11:12:39
Part 2:
议用法:

对于 su 的传统的用法是允许 superuser 不经输入密码的变更为其它次要的用户的身份. 为在 Linux-pam 里实现这种行为, 下面两行需要加入设定档:

```
#
# su authentication. Root is granted access by default.
#
su auth sufficient pam_rootok.so
su auth required pam_unix_auth.so
```

注意. 对于以 superuser 执行(或者在系统开机时被开启)的程序, 应该不要用此模块去认证用户.

6.23 The securetty module
概要

模块名:

pam_securetty
作者:

Elliot Lee <sopwith@cuc.edu>
维护者:

Red Hat Software:
currently Michael K. Johnson <johnsonm@redhat.com>
(if unavailable, contact Elliot Lee <sopwith@cuc.edu>).
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

/etc/securetty file
网络接口:

为使此有意义的, 需要应用程序设定正确的 pam_TTY.

概述

提供标准的 UNIX 的安全的 tty 的认证.
Authentication component

识别的参数:
描述:

提供标准的 Unix 安全 tty 的检查， 除非 pam_TTY 的值列在 /etc/securetty 里, 对 root 的认证将会失败. 对所有其他用户, 则为成功.
范本/建议用法:

规范的用法, 这模块应该作为 required 的认证方法出现在任何 sufficient 的认证之前.

6.24 Time control
概要

模块名:

pam_time
作者:

Andrew G. Morgan <morgan@parc.power.net>
维护者:

Author
提供的管理组:

account
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

需要一个设定档:/etc/security/time.conf
网络接口:

仅通过 pam_TTY

概述

运行一个规范得很好的系统偶尔也会包含对特定服务的有选择的限制访问. 此模块提供了对系统服务的时间上的访问控制. 它的行为由一个设定档决定. 此模块可以设定成基于用户名, 时间, 星期, 具体服务和请求服务时的终端名的对(单个)用户的拒绝访问.
Account component

识别的参数:
描述:

此模块基于设定档中的制定的规则进行动作:/etc/security/time.conf. m 每个规则有如下格式,

services;ttys;users;times

每个规则占一行, 结束于换行符或者注解的开始:`#'. 由分号`;' 分隔成四个栏位. 这些栏位是:

* services - 由此规则影响的服务名的逻辑列.
* ttys - 指示受此规则保护的终端名字的逻辑列表逻辑列表.
* user - 此规则作用到的用户的逻辑列表.

逻辑列表意味一串符号(和适当的 pam_相关), 包含不超过一个的通配符:`*', 和可选的非操作符的前置:`!'. 这一串表达式由两个逻辑操作符连接:& (逻辑与)和| 逻辑或). 两个例子:!morgan&!root,

表示此规则不会应用到用户 morgan 和 root;tty*&!ttyp*, 表示此规则仅对控制台终端而不是伪终端适用.
* times - 此规则的适用时间的逻辑列表. 每个元件的格式是周/时间-范围. 周由两个字符指定.比如, MoTuSa, 表示周一周二和周六. 注意重复的周被 复位; MoTuMo 表示周二, MoWk 意为除了周一的所有工作日. 两个字符的组合可以是,

Mo Tu We Th Fr Sa Su Wk Wd Al

最后两个词是周末和 每周的整个 7 天.

时间范围部分是一对 24 时制的时间, HHMM, 以一个连字符隔开 -- 表示开始和结束的时间. 如果结束时间早于开始时间, 就假定是到第二天的这个结束时间. 例如, Mo1800-0300 表示允许的时间是从周一的晚上 6 点到第二天早上 3 点.

注意, 时间的限制只有在当前 3 个栏位都满足用户请求服务的应用时才适用.

为了便利和可读性, 一条规则可以由\换行符'结尾以超过一行.
范本/建议用法:

要使用此模块可以在 Linux-pam 的设定档里加入下面的行:

```
#
# apply pam_time accounting to login requests
#
login account required pam_time.so
```

这里, 我们把这模块应用在 login 的服务.

可以放在/etc/security/time.conf 里的规则举例如下:

```
login ; tty* & ; !ttyp* ; !root ; !Al0000-2400
```

除了 root,禁止所有用户在任何时间从控制台登录.

```
games ; * ; !waster ; Wd0000-2400 | Wk1800-0800
```

games (使用 Linux-pam 的某游戏) 只在非工作时间可以访问. 此规则不限制用户 waster.

注意, 目前还没有一个后台进程去强制终止一个超过时间范围的会话. 这要有个对策.

错误格式的规则会作为错误记录在 syslog(3).

6.25 The Unix Password module

概要

模块名:

pam_unix
作者:
维护者:

作者.
提供的管理组:

account; authentication; password; session
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:
网络接口:

概述

这是标准 Unix 的认证模块. 它从系统库呼叫标准的调用来获取和设定帐号信息以及认证信息. 通常这些从/etc/passwd 和/etc/shadow 文件(如果开启了 shadow 功能)获得.
Account component

识别的参数:

debug; audit
描述:

参数 debug 使此模块的帐户相关的函数 syslog(3)更多的它的动作的信息 (属于此模块的其它功能的参数会被直接忽略掉, 但是别的参数会当成错误而被 syslog(3)). 参数 audit 会丢出更多的信息.

基于以下的 shadow 元素: expire; last_change; max_change; min_change; warn_change, 此模块执行确认帐号和密码状态的工作. 至於后面, 它可以建议用户改变密码或, 通过返回 pam_AUTHTOKEN_REQD, 延迟给用户提供服务直到其生成一个新密码. 列在上面的那些元素在 GNU Libc 的 info 文档里有解释. 如果用户的记录里没有包含一个或者更多的这些元素, 相对的 shadow 检查将不被执行.
范本/建议用法:

在帐号管理模式下, 可以这样来安装此模块:

#

Ensure users account and password are still active
#
login account required pam_unix.so

Authentication component

识别的参数:

debug; audit; use_first_pass; try_first_pass; nullok; nodelay
描述:

参数 debug 使此模块的帐户相关的函数 syslog(3)更多的它的动作的信息. 参数 audit 会丢出更多的信息.

此模块的默认设定是如果正式的密码为空则不允许用户访问服务. 参数 nullok 用来使此默认设定无效.

当给出参数 try_first_pass, 在提示用户输入密码之前, 模块会拿通过之前堆叠的 auth 模块认证的密码来试. 参数 use_first_pass 迫使模块使用上述的密码并且永远不会提示用户 - 如果没有可用的密码或密码不能通过认证, 用户将被禁止访问.

参数 nodelay 用来阻止认证部件在认证失败时的延时请求. 默认动作是请求一个一秒钟以上的失败-延迟.

属于此模块的其它功能的参数会被直接忽略掉, 但是别的参数会当成错误而被 syslog(3).

一个帮手程序, unix_chkpwd, 在当用户的密码报存在一个读保护的数据库里时提供对用户密码的检查. 这个程序非常简单, 它仅仅检查呼叫它的用户的密码. 它被此模块的认证部件以用户的名义来透明的呼叫. 如此象 xlock 这样的程序就不需被设成 setuid-root 而可以工作.
范本/建议用法:

此模块的正确功能可以由一个适当的/etc/nsswitch.conf 来支配, 那里指定的用户数据库确定用来认证的用户记录的来源.

在帐号管理模式下, 可以这样来安装此模块:

#
Authenticate the user
#
login auth required pam_unix.so

Password component

识别的参数:

debug; audit; nullok; not_set_pass; use_authtok; try_first_pass; use_first_pass; md5; bigcrypt; shadow; nis; min; max; obscure; remember
描述:

pam_unix 模块的这一部分执行更新用户密码的任务.

对于传统的 UNIX 用户数据库(同时保存用户密码), 参数 md5 用来采用 MD5 的加密法以对抗不安全的传统的 crypt(3)调用. 还有种选择, 参数 bigcrypt 采用 DEC(数字设备公司)的对标准 UNIX 的 crypt()的算法进行 `C2'扩展的算法以加密超过 8 位长的密码.

参数 nullok 用来允许改变一个原先为空的密码. 如果没有此参数, 空的密码会当成是被锁住的帐号.

参数 use_first_pass 用来阻止选择旧的和新的密码,而是采用 password 里此模块之前的模块所保留的密码. 参数 try_first_pass 用来防止当 pam_pwdb 跟随另一个 password 模块时用户重复输入也许是共享的旧密码 - 如果旧密码不正确, 再提示用户输入. 参数 use_authok 用来迫使 此模块设定新密码时采用 password 堆叠里先前的模块提供的值. (这用法在先前介绍的 Cracklib 一节里有个范本).

参数 not_set_pass 用来告诉模块不去关心给/从其它(堆叠的)密码模块提供/获取可用的旧密码或新密码.

参数 debug 使此模块的密码相关的函数 syslog(3)更多的它的动作的信息. 其它参数会被当成错误信息记录在 syslog(3). 参数 audit 会丢出更多的信息.

通过设定 nis 参数, pam_unix 会试图采用 NIS RPC 来设定新密码.

参数 remember 带有一个值. 这值是为每个用户保存的最近使用的密码的个数. 它们被保存在 /etc/security/opasswd 以防止用户频繁的交替使用同样的一组密码.

参数 min 和 max 允许控制密码的常度. 有个固定的默认值是 1 到 8. 这些值本身算在内.

参数 obscure 允许一些额外的密码检查. 这效仿原始的 shadow 包里的含糊检查, 这过程和 pam_cracklib 模块非常相像(不做字典检查), 它执行以下一些检查:

* Palindrome - 新密码是否为旧密码的回文? 回文是指顺着读和反着读都一样的词(比如 madam 和 radar).
* Case Change Only - 新密码是否只是拿旧密码改了些大小写?
* Similar - 新密码是否太像旧密码?

* Simple - 新密码是否太简单? 这决定于密码的长度和采用不同种字符的个数(比如,字母, 数字...).
* Rotated - 新密码是否为旧密码的轮转?(比如,"billy"和"illyb")

范本/建议用法:

标准用法:

```
#
# Change the users password
#
passwd password required pam_unix.so
```

和可插入式密码检查模块,pam_cracklib 关联堆叠的例子:

```
#
# Change the users password
#
passwd password required pam_cracklib.so retry=3 minlen=6 difok=3
passwd password required pam_unix.so use_authtok nullok md5
```

Session component

识别的参数:
描述:

模块的此部件没有参数. 它的作用就是在用户会话开始和结束时往 syslog(3) 里记录用户名和服务类别.
范本/建议用法:

session 部件的用法是单纯的:

```
#
# session opening and closing
#
login session required pam_unix.so
```

6.26 The userdb module
概要

模块名:

pam_userdb
作者:

Cristian Gafton <gafton@redhat.com>
维护者:

作者.
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:

需要有 Berkeley DB.
网络接口:

概述

在一个.db 数据库里查找用户并且与报存在这数据库中的密码进行比对.
Authentication component

识别的参数:

debug; icase; dump; db=XXXX;
描述:

此模块用来与储存在 BerkeleyDB 数据库里的资料对用户/密码进行验证. 数据库以用户名为索引, 和用户名对应的栏位是以明码形式保存的密码, so caution must be exercised over the access rights to the DB database itself.. 模块以交互机制从用户端获取输入的密码. 如果此密码用在其他认证模块(象 pam_unix)的前面, 那么你得要告诉那些模块从 pam_AUTHTOK(由此模块设定)处读取输入的密码.

模块的默认动作可以由以下的参数改变:

* debug - 给 syslog(3)提供更多的调信息.
* icase - 忽略大小写的密码比对.
* dump - 往 log 里下载数据库里所有的的记录.(呀, 默认情况下不要这样做!)
* db=XXXX - 使用 XXXX 这个数据库文件. 注意, Berkeley DB 通常会在文件名后面加上扩展名, 所以你得像 /etc/foodata 这样指定文件而不是/etc/foodata.db.

范本/建议用法:

这是一个 ftp 服务的设定档(通常是 /etc/pam.d/ftp), 会接受用户名/密码对存放在/tmp/dbtest.db 中的用户登入.

#%pam-1.0
auth required pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed
auth sufficient pam_userdb.so icase db=/tmp/dbtest
auth required pam_unix.so shadow nullok try_first_pass
auth required pam_shells.so
account required pam_unix.so
session required pam_unix.so

6.27 Warning logger module
概要

模块名:

pam_warn
作者:

Andrew G. Morgan <morgan@parc.power.net>
维护者:

作者.
提供的管理组:

authentication; password
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:
网络接口:

记录远程用户和远程主机的信息(如果可以知道相关的 pam-items)

概述

此模块主要用来记录试图进行认证或者更新密码的动作.
Authentication+Password component

识别的参数:

描述:

往 syslog(3)里记录服务名,终端名,用户名,远端用户名和远程主机. 不是所有的项目都能侦测到, but instead obtained from the standard pam-items.
范本/建议用法:

在设定档一节 above 有一个范本 .

6.28 The wheel module
概要

模块名:

pam_wheel
作者:

Cristian Gafton <gafton@redhat.com>
维护者:

作者.
提供的管理组:

authentication
Cryptographically sensitive:
安全等级:
代码清洁度:
系统依赖:
网络接口:

概述

只给属于 whell(gid=0)群组的用户授予 root 的访问权限.
Authentication component

识别的参数:

debug; trust; deny; group=XXXX
描述:

此模块用来实现所谓的重要人物 群组. 默认设定时, 如果请求的用户是 wheel 的组成员就赋予其 root 的的访问权限. (首先, 模块检查组 wheel 是否存在. 如果不存在,那么 id=0 的组被当作 wheel.

模块的默认动作可以通过在/etc/pam.conf 里设定以下参数来改变:

* debug - 提供更多的调试信息给 syslog(3).
* trust - 此选项指示模块当一个请求 root 权限的用户是 wheel 组的成员时返回 pam_SUCCESS. 这种情况的默认动作是返回 pam_IGNORE . 使用 trust 选项可以使 wheel 的组成员未经输入密码就能成为 root. 小心使用.
* deny - 这选项用来推翻模块的行为逻辑. 如果一个属于 wheel 组的用户试图得到 uid=0 的权限时, 就禁止访问(wheel 这个词也许就变得毫无意义!): 其目的是为了接下来的 group= 参数而为.
* group=XXXX - 代替对 gid=0 的组的眷顾, 而使用用户自己定意 XXXX 组. 这里的 XXXX 是组的名字而不是组的 ID.

范本/建议用法:

为把能取得 superuser 权限的用户限制在 wheel 里, 使用下面的设定:

```
#
# 默认给 root 赋权限(rootok), 只有 wheel 的组成员有可能成为 root( wheel), 但仍然必须
# 通过 unix_auth 的认证.
#
#
su auth sufficient pam_rootok.so
su auth required pam_wheel.so
su auth required pam_unix_auth.so
```

7. 文件

/usr/lib/libpam.so.*

提供给应用程序访问 Linux-pam API 的共享库.
/etc/pam.conf

Linux-pam 的设定档.
/usr/lib/security/pam_*.so

Linux-pam 的动态可装载目标文件(模块)的主要所在位置.

804 ☆PAM API

1、框架 API：
任何一个支持 pam 的服务程序在进行认证时必须以 pam_start( )开始进行初始化，最后以 pam_end( )结束以便进行清理工作。

2、认证管理 API：
pam_authenticate( )对用户名/密码进行认证。
pam_setcred( )用来修改用户的秘密信息。

3、帐户管理 API：
pam_acct_mgmt( )检查帐户本身是否有权限登录系统、帐户是否过期、帐户是否有登录时间限制等。

4、密码管理 API：
pam_chauthtok( )修改用户的密码。

5、会话管理 API：
一个会话以 pam_open_session( )开始，最后以 pam_close_session( )结束。

6、其它：
pam_get_item( )、pam_set_item( )用来读写 pam 事务(transaction)的状态信息。
pam_get_data( )、pam_set_data( )用来取得和设置 pam 模块及会话的相关信息。
pam_putenv( )、pam_getenv( )、pam_getenvlist( )用来读写环境变量。
pam_strerror( )返回相关的错误信息。

805 ☆PAM SPI

当服务程序（ftpd、telnetd 等）调用 pam API 函数 pam_xxx( )时，由 pam 框架（libpam）根据该服务在/etc/pam.conf 文件中的配置调用指定的 pam 模块中对应的 SPI 函数 pam_sm_xxx()。如下：

API 函数的名字为 pam_xxx( )，对应的 SPI 函数的名字为 pam_sm_xxx( )，即每个服务模块需要引出相应的函数以供 libpam 调用。为方便对照，再列一下。

API 对应的 SPI
帐号管理 pam_acct_mgmt( ) pam_sm_acct_mgmt( )
认证管理 pam_authenticate( ) pam_ sm_authenticate( )
密码管理 pam_chauthtok( ) pam_ sm_chauthtok( )
会话管理 pam_open_session( ) pam_ sm_open_session( )
会话管理 pam_close_session( ) pam_ sm_close_session( )
认证管理 pam_setcred( ) pam_ sm_setcred( )

806 ☆ldd

怎么知道程序是否已经含有 pam 的代码了呢?(但不总是可靠)
ldd <程序>

或者检查/etc/pam.d 目录中的配置文件

807☆system-auth

```
auth            required        /lib/security/$ISA/pam_env.so
auth            sufficient      /lib/security/$ISA/pam_unix.so likeauth nullok
auth            required        /lib/security/$ISA/pam_deny.so

account         required        /lib/security/$ISA/pam_unix.so
account         sufficient      /lib/security/$ISA/pam_succeed_if.so uid < 100 quiet
account         required        /lib/security/$ISA/pam_permit.so

password        requisite       /lib/security/$ISA/pam_cracklib.so retry=3
password        sufficient      /lib/security/$ISA/pam_unix.so nullok use_authtok md5 shadow
password        required        /lib/security/$ISA/pam_deny.so

session         required        /lib/security/$ISA/pam_limits.so
session         required        /lib/security/$ISA/pam_unix.so
```

system-auth 文件为每一个模块设置了最小要求检测
authconfig 在每一次运行后都重写/etc/pam.d/system-auth 文件,所以你不应该直接修改这个文件.(如用 vi 编辑它).

通过编辑 /etc/pam.d/system-auth 来限制所有服务，把以下这行添加到以 auth 开头的所有行后
account required /lib/security/$ISA/pam_access.so

编辑 /etc/security/access.conf 限定 finance 组的用户只能在第二个虚拟控制台登陆。为了达到这个目的，在这个文件的最后一行添加：
- : finance : ALL EXCEPT tty2

808☆access.conf

/etc/security/access.conf 配置文件说明：
* 该文件的每一行由如下三个字段构成，中间使用冒号分割：
权限 : 用户 : 来源
* 权限字段可以是”+”(即允许访问),”-”(禁止访问);
* 用户字段可以是用户名、组名以及诸如 user@host 格式的用户名，ALL 表示任何人，具有多个值时，可以用空格分开。
* 来源字段可以是 tty 名称（本地登录时）、主机名、域名（以”.”开始）,主机 ip 地址，网络号（以”.”结束）。ALL 表示任何主机，LOCAL 表示本地登录。
* 可以使用 EXCEPT 操作符来表示除了…之外。

所以：

* +:root:LOCAL---表示 root 用户可以从本地登录。
* +:bye2000 tom:192.168.1.---表示 bye2000 和 tom 可以从 192.168.1.0/24 网段 telnet 登录。
* -:ALL:ALL---表示拒绝其他任何人登录。

account required /lib/security/pam_access.so accessfile=/etc/security/access.conf

809 ☆限制 NIS 用户

我们的客户端现在是公司 NIS 体系的一部分，因为他储存了秘密数据，不是所有的用户都可以访问这台机器，只有特定的远程用户才能访问。

1. 这个测试需要添加一个 NIS 用户，使用 useradd 命令添加一个名叫 baduser 的用户。

useradd -u 1026 baduser
passwd baduser

2. 一个解决方案是使用 pam_listfile，只允许 nisuser 访问我们的系统。打开 /etc/pam.d/system-auth ，紧接着 auth 开头的之后添加以下一行:

account required /lib/security/pam_listfile.so item=user sense=allow
file=/etc/nisusers onerr=fail

3. 假如测试目前的设置，你将会发现连 root 也不能登陆，所以千万不要关掉 root 的 shell！你要创建 /etc/nisusers 然后把所有允许访问的用户添加到文件中，一行一个用户名，我们只想允许 nisuser 用户，所以我们的文件会非常短。

4. 现在如果你想登录到文本控制台，只有 nisuser 可以进入，因为其他人不在文件中，把 root 添加到 /etc/nisusers 中。

5. 我们的任务还是允许所有本地用户登陆的，我们可以把 passwd 文件中的用户都添加到我们的列表中，但这不是最好的方法，我们可以使用 PAM 库中的 pam_localuser 来达到目的。添加以下这行到 pam_localuser.so 之后。

account required /lib/security/pam_localuser.so

6. 测试这样的配置，你会发现仍然只有 root 可以登陆，为什么呢？

7. 是因为 required 字段的关系，把上面添加的两行的 required 都改成 sufficient，现在好了吧？如果改成 requisite 会怎么样？

8. 清理:再次运行 authconfig 工具，删除所有设置，并且禁用 NIS。

810 ☆定位易被攻击的文件或目录

查找有 SUID 和 SGID 的文件，并且把他们的名字存在 /root/stickyfiles 中:
find / -type f -perm +6000 2> /dev/null > /root/stickyfiles

查找任何人都可以写入的文件，把它们的名字储存在 /root/worls.writalbe.files:
find / -type f -perm -2 > /root/world.writalbe.files

811 ☆tmpwatch

清理临时文件目录

运行 tmpwatch 的 test 选项，这样可以看一下哪些文件 7 天没有人访问了:
tmpwatch -v --test 168 /tmp

812 ☆日志

日志文件所处的位置
日志文件所处的位置都在/var/log 目录下，前提是您没有对日志配置文件/etc/syslog.conf 进行过特别的配制。

有个查看日志的工具
#system-logviewer

如果是看启动时的日志，应该是
/var/log/boot.log
记录服务的启动和停止

日志文件的配置文件
日志文件的配制文件，在/etc/syslog.conf，如果我们要修改日志配制文件，我们要首先要备份。这一点，是我们进行系统管理的首要任务。

Log all kernel messages to the console.
Logging much else clutters up the screen.
#kern.* /dev/console

Log anything (except mail) of level info or higher.
Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none /var/log/message
s

The authpriv file has restricted access.
authpriv.* /var/log/secure 安全验证日志，系统生成的日志文件是放在了/var/log/secure

Log all the mail messages in one place.
mail.* /var/log/maillog 这个是电子邮件系统的功能，这个日志文件是在/var/log/maillog 目录下。

Log cron stuff
cron.* /var/log/cron 这个是定时作业时信息

Everybody gets emergency messages
*.emerg * 这是 syslog 对日志所设置的级别,emerg 表示系统已经不可用

Save news errors of level crit and higher in a special file.
uucp,news.crit /var/log/spooler 这是 syslog 对 news 和 uucp 的日志所设置的级别,crit 表示危急，但事故还没有发生，将要发生。

Save boot messages also to boot.log
local7.* /var/log/boot.log 开机系统日志，用 local7 来表示，日志文件的位置处在/var/log，日志文件是 boot.log

#
INN
#
news.=crit /var/log/news/news.crit
news.=err /var/log/news/news.err
news.notice /var/log/news/news.notice

日志类型
authpriv 安全性/验证的信息，通过这个，我们可以查看比如 telnet 和 ssh 之类登入系统方面的日志。这对于防黑有重要作用，不可小视。
cron 任务调度信息，有点象 windows 中的计划任务，我们可以通过这个程序在什么时间做什么事。他的配制文件在 /etc/crontab 中，在这里我们是说它的日志文件的配制
kern 这是系统内核的日志，这个要我们自己定义存放位置，我们可以在/etc/syslog.conf 中自己来定义存放位置。比如，我们可以在 syslog.conf 中加一行，比如是这样的
ker.debug /var/log/kern.log
local0-local7 自定义级别，开机系统日志，用 local7 来表示，日志文件的位置处在/var/log，日志文件是 boot.log
lpr 看名字也应该知道，这是打印的日志文件，这个我们也一样可以自己来定义。在下面，我们再逐步深入说一下如何写系统日志
mail 是电子邮件的，sendmail,qmail 等信息

news 是新闻组服务器的。
user 一般和户信息
syslog 内部 log 信息
auth 也是用户登入的信息，安全性和验证性的日志
uucp 全称是 UNIX-TO-UNIX COPY PROTOCOL 的信息

日志级别
日志系统管理员来维护系统的，系统日志的内容太多，所以就有必要把日志按级别来排序，这样能方便管理员发现比较紧急和重要的问题，以着手处理和解决。
这里有一个主次顺序，也就是重要的都放在前面，级别是由高而低的。
emerg 系统已经不可用，级别为紧急
alert 警报，需要立即处理和解决
crit 既将发生，得需要预防。事件就要发生
warnig 警告。
err 错误信息，普通的错误信息
notice 提醒信息，很重要的信息
info 通知信息，属于一般信息
debug 这是调试类信息
* 记录所有的信息，并发到所给所有的用户，以上的信

日志设置或者语法格式的书写
在/etc/syslog.conf 中，根据我们自己的情况，可以配制或者定义日志文件。语法格式如下，也比较简单。。

日志类型.等级 日志存放位置[要用绝对路径]

举个例子来说
kern.debug /var/log/kern.log

看到如下的一行。这代表什么意思呢。
authpriv.* /var/log/secure
这个代表的意思是：所有验证类级别的日志都存放在 secure 这个日志文件里。有时，我们也会在/var/log 目录里，看到 secure1 之类的，其实也是这类的日志，我们要灵活一下。是不是？
通过这个文件，我们可以看到验证类的日志，比如 telnet 和 ssh 等。如果别人用 telnet 我们的机器，我们就要查看这个文件了。我们可以通过
#more secure | grep telnet 来看，当然用 more 也能一页一页的看过去，我的目的仅仅是想知道是不是这个文件能看到这方面的东西，比如我用机其它器 telnet，我的

813☆utmpdump

/var/log/wtmp
所有成功的登录记录在其中。这个文件将无限制地增大，所以必须有规律的清除

/var/run/utmp
当前登录着的用户列在其中。这个文件直到系统下次启动或关机前有效。

utmpdump wtmp
翻译

814 ☆远程日志

syslogd -m 0 -r
启动就可以了,-r 很重要只能在最后面,后面也一样,否则不能成功

配置接收远程的消息
编辑 /etc/sysconfig/syslog :
SYSLOGD_OPTIONS="-m 0 -r"
shu 后 service syslog restart

配置发送给远端
/etc/rc.d/init.d/syslog 中
SYSLOGD_OPTIONS="-m 0"
改为
SYSLOGD_OPTIONS="-m 0 -r"
然后 service syslog restart

iptables -I INPUT -p udp --dport 514 -j ACCEPT

配置 syslogd 发消息给别的机器,/etc/syslog.conf 中改为
. @192.168.152.128
user.* @panda

使用 logger 创建一个 syslog 的消息
logger -i -t yourname "this is a test"
这则消息是否显示在你旁边机器的 /var/log/messages 中了呢？