

《实战Linux Shell编程与服务器》

图书基本信息

书名：《实战Linux Shell编程与服务器管理》

13位ISBN编号：9787121103285

10位ISBN编号：7121103281

出版时间：2010-3

出版社：电子工业出版社

作者：卧龙小三

页数：533

译者：梁昌泰

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

前言

2002年，我曾在台南县对全县中、小学校的网管讲授Shell程序设计课程。当时关于Shell的书籍和参考资料并不多（至今仍然如此），只好自编一份入门讲义，供上课的学员参考。课程结束后，我把它放到网络上，让其他同学也能参考。没想到，这份讲义，由于写得十分清楚、易懂，受到许多人的欢迎。这几年来，陆续接到很多网络上朋友的来信，希望我能进一步把Shell程序设计的入门讲义写成更详细的工具书，不过，由于个人工作上的关系，一直没有实现。今年，机缘巧合，各方面的条件和时机恰好配合，所以，这本书就出现在这里啦！许多人可能不了解，Shell其实是操作系统的重要组成部分之一。当我们管理一台主机时，在系统核心和用户之间的接口就是Shell，我们几乎必须时刻面对它。Shell能解译使用者输入的命令并执行命令。这些命令本身，通常用来访问文件系统、建立行程、管理行程、操作输出设备、管理内存，以及操作和网络有关的应用。因此，对Shell的操控能力，可以说大约就等于是主机的管理能力，所以，想要对主机管理有更深一层认识的人，不可不熟悉Shell。

笔者常说：凡是交给计算机做的事，不要由人来做（因为，人会累，计算机不会累）。主机管理的上乘之道，就在于能够充分实践这句话。换言之，就是把工作予以自动化。那么，主机管理者如何把工作自动化呢？关键就在于他是否具备Shell程序设计的能力。我们可以说：对Shell的操控能力和Shell程序设计的能力，是每一位主机管理者应具备的基本知能。唯有如此，这样的管理人员才具备强大的生产力（有生产力的管理者，才能偷懒，有时间做自己喜欢做的事，比如：思考如何创新）。

这几年来，由于自由软件运动蓬勃发展，一般人接触Linux / BSD等系统的机会多了，较几年前来说，真是不可同日而语。但许多人在登录主机之后，却往往不知道要做什么，或如何运用它来解决问题，实在是很可惜。本书希望能带领读者了解Shell的重要性，熟悉Shell的运用方法，建立自动化管理主机的能力，迅速、轻松地解决问题。

《实战Linux Shell编程与服务器》

内容概要

《实战Linux Shell编程与服务器管理》内容简介：操作系统Shell编程一直是系统管理员必备高级技能之一。主机管理的上乘之道，就在于尽量让计算机去完成所有的琐事。要达到这个境界，管理者必须具备相当程度的Shell操控能力与Shell程序设计能力，笔者拥有丰富的主机管理实务经验，以及多年的教学经验，能引领读者轻松跨越Bash Shell程序设计的门槛，摆脱菜鸟劳碌的命运，达到轻松管理，主机不出包的境界。

《实战Linux Shell编程与服务器管理》完整涵盖Bash 3.x的各项功能，并介绍最新Bash 4.0的重要功能；由简入繁、循序渐进，建立扎实的Bash Shell程序设计基础；各章提供许多范例，充分展示Bash Shell程序设计的技巧；带领读者学习如何设计自动化程序，轻松解决问题，增进工作效率；还包含许多管理实务的技巧，可快速提升读者主机管理的能力。

《实战Linux Shell编程与服务器管理》适合系统管理员、网络管理员、架构师及普通学习者参考学习。

Chapter 1 Shell简介	1.1 Shell是什么	21	2 Shell的历史简介	41	3 Bash Shell的功能	51	4 第一个Bash Shell程序：“哈啰！Bash Shell！”	61	5 第一个有用的Bash Shell	7																																																						
Chapter 2 布置Bash Shell的环境	2.1 在Linux平台布置Bash Shell的环境	102	2.2 升级B2D的Bash版本	102	3 自行编译最新版的Bash	112	4 切换使用新版的Bash Shell	13	5 chsh的用法	132	6 在FreeBSD平台布置Bash Shell的环境	152	7 在OpenBSD平台布置Bash Shell的环境	162	8 在Windows平台布置Bash Shell的环境	18																																																
Chapter 3 基础概念介绍	3.1 登录、注销	22	2 登录主机	22	3 注销主机、结束终端程序	23	4 文件、目录	23	5 文件	23	6 目录	26	7 3 通配符与转义字符	30	8 通配符	30	9 转义字符	31	10 4 字符集合	33	11 5 括号扩展	34	12 6 系统默认开启的文件	35	13 7 标准输入/输出转向	35	14 8 转向输出	36	15 9 转向附加	36	16 10 转向输入	37	17 11 转向输入和转向输出合用	37	18 12 利用转向输出做简易编辑	38	19 8 管道	38	20 9 前台工作、后台工作	40																								
Chapter 4 Bash Shell程序的结构	4.1 Shell程序的组成	42	4.2 设定执行权限	44	4.3 执行Bash Script的方法	44	4.4 Bash Script排错的方法	45	4.5 Bash Script执行的原理	47																																																						
login Shell	47	4.6 Bash Shell的启动配置文件	50	5 Bash不同的运行模式	50																																																											
Chapter 5 基本操作介绍	5.1 内置命令	56	5.2 命令行程序	71	5.3 执行多个命令的方法	88	5.4 记录命令的执行过程	90	5.5 命令行编辑的方法	91	5.6 Bash分析命令行的方式	98																																																				
Chapter 6 变量与字符串操作	6.1 变量是什么	102	6.2 变量的命名	103	6.3 设定变量	104	6.4 取得变量值	105	6.5 取消与清空变量	109	6.6 变量和引号	110	6.7 变量的有效范围	112	6.8 环境变量	113	6.9 输出变量	113	6.10 取消环境变量值	115	6.11 Bash的内置变量	115	6.12 调整变量的属性	125	6.13 只读变量：使用readonly命令	125	6.14 只读变量：使用declare命令	126	6.15 调整变量的属性	127	6.16 11 别名	128	6.17 建立别名	128	6.18 替换别名	129	6.19 取消别名	130	6.20 12 自定义环境	130	6.21 以一般账号的角色工作时	130	6.22 以系统管理员的角色工作时	133	6.23 13 数组	133	6.24 数组的用法	134	6.25 取出数组所有的元素	135	6.26 取得数组元素的个数	135	6.27 取消数组或数组元素	136	6.28 14 Here Document	136	6.29 用Here Document设定变量	136	6.30 控制Here Document的格式	138	6.31 利用Here Document做多行批注	139	6.32 利用Here Document，打包C（或其他程序语言）的原始码	140
Chapter 7 高级变量	7.1 变量扩展：测试存在性及空值	144	7.2 “变量存在”的定义	144	7.3 测试变量“存在与否”的基本用法	145	7.4 测试变量“不存在”或其值为空：“传回”一个默认值	146	7.5 测试变量“不存在”或其值为空：给变量设一个默认值	148	7.6 测试变量是否“不存在”或其值为空：提示错误信息	149	7.7 测试变量的“存在性”	151	7.8 2 变量扩展：取字符串切片、字符串长度	152	7.9 取字符串切片	152	7.10 计算字符串长度	155	7.11 3 变量扩展：对比样式	156	7.12 由字符串前面对比，删除相符者	156	7.13 由字符串后面对比，删除相符者	158	7.14 取代或删除部分字符串	159	7.15 把对比符合的字符串删除	160	7.16 要求样式在句首或句尾	160	7.17 4 变量扩展：取变量名称列表、数组索引列表	162	7.18 取变量名称列表	162	7.19 取数组索引列表	162	7.20 2.5 命令替换	163	7.21 6 算术扩展	165																						
Chapter 8 算术运算	8.1 简介	168	8.2 算术扩展	169	8.3 使用外部程序expr做算术运算	174	8.4 使用\$[]做算术运算	177	8.5 使用内置命令declare、let做算术运算	178	8.6 declare的用法	178	8.7 let的用法	179																																																		
Chapter 9 流程控制	9.1 命令的结束状态	182	9.2 if条件判断	183	9.3 简易的if语法：if-then-else	183	9.4 if的完整语法	184	9.5 条件测试的写法	187	9.6 条件判断式的真假值	195	9.7 关于文件属性的判断式	195	9.8 关于字符串的条件判断式	197	9.9 关于算式的条件判断式	199	9.10 关于Bash选项的条件判断式	200	9.11 5 case条件判断	200	9.12 case的语法	200	9.13 高级样式	202	9.14 6 for循环	204	9.15 for的语法	204	9.16 for无穷循环	207	9.17 7 while循环	207	9.18 while的语法	207	9.19 使用while循环，读取文件内容	209	9.20 while无穷循环	210	9.21 8 until循环	211	9.22 until的语法	211	9.23 until无穷循环	212	9.24 9 select命令	212	9.25 10 break和continue	214	9.26 break命令	214	9.27 continue命令	216	9.28 11 综合运用	216								
Chapter 10 函数	10.1 函数的用法	220	10.2 函数的语法	220	10.3 函数的结束状态	222	10.4 函数与变量的作用范围	224	10.5 函数的作用范围	224	10.6 变量的作用范围	225	10.7 3 位置参数	226	10.8 传递参数	226	10.9 命令行的位置参数	229	10.10 移动位置参数	229	10.11 指定位置参数值	230	10.12 取用命令行的选项和参数	233	10.13 4 建立函数库	235	10.14 5 递归函数	237	10.15 6 实例应用	238																																		
Chapter 11 转向	11.1 文件代码	244	11.2 操作文件	244	11.3 开启文件	244	11.4 关闭文件	245	11.5 复制文件代码，建立文件连接	245	11.6 3 转向的用法	247	11.7 转向输入	247	11.8 转向输出	248	11.9 转向附加	250	11.10 标准错误伴随标准输出做转向	250	11.11 Here Document转向	251																																										
Chapter 12 trap——陷阱触发	12.1 信号（signal）	254	12.2 进程编号PID	254	12.3 信号列表	255	12.4 传送信号的方法	256	12.5 常用的信号	257	12.6 2 trap的运用	260	12.7 trap的语法1	260	12.8 trap的语法2	261	12.9 trap的语法3	261	12.10 trap的语法4	262																																												
Chapter 13 sed和awk入门	13.1 正则表达式	264	13.2 何谓正则表达式	264	13.3 sed的用法	266	13.4 实例应用	268	13.5 awk的用法	271	13.6 实例应用	272																																																				
Chapter 14 进程管理和工作控制	14.1 进程管理	276	14.2 ps命令	276	14.3 top命令	281	14.4 http命令	283	14.5 传送信号	284	14.6 调整进程的优先级	284	14.7 nice命令	285	14.8 renice命令	286	14.9 PRI和NI	287	14.10 14.2 工作控制	287	14.11 以实例说明工作控制	287	14.12 常用的工作控制命令	290	14.13 3 进程替换	292																																						
Chapter 15 历史指令	15.1 历史指令的功能	296	15.2 history命令	297	15.3 fc命令	298	15.4 5.2 历史指令扩展	298																																																								
Chapter 16 使用Shell Script撰写文字和图形接口程序	16.1 使用dialog撰写文本接口程序	306	16.2 是或否对话框（yesno）	309	16.3 多选一对话框（radiolist）	310	16.4 选单对																																																									

对话框 (menu) 312 单行输入对话框 (inputbox) 314 输入密码对话框 (passwordbox) 315 显示进度对话框 (gauge) 317 核查列表对话框 (checklist) 320 文件对话框 (textbox) 321 信息对话框 (infobox) 321 调整时间对话框 (timebox) 322 倒计时对话框 (pause) 322 选择文本区对话框 (fselect) 322 16.2 使用xdialog撰写图形接口程序 323 显示信息 (msgbox) 323 多选一对话框 (radiolist) 324 单行输入对话框 (inputbox) 325 双行输入对话框 (2inputsbox) 326 三行输入对话框 (3inputsbox) 328 下拉列表对话框 (combobox) 329 重设大小范围的对话框 (rangebox) 329 两个滚动条的对话框 330 3个滚动条的对话框 330 spinbox对话框 331 编辑对话框 (editbox) 331

Chapter 17 文件操作 17.1 取得目录内容：文件列表 334 利用ls取得目录内容 334 17.2 判断文件是否存在 336 17.3 建立文件 338 建立一般文件 338 建立目录 340 建立区块设备文件 341 建立字符设备文件 343 建立FIFO文件 343 建立Socket文件 345 建立符号链接文件 345 17.4 取得文件的属性 346 17.5 对比文件的差异 352 diff的其他用法 355 17.6 搜寻文件内容 358 17.7 建立临时文件 361 17.8 复制文件 363 17.9 大、小写文件名互换 366 17.10 转换编码 371 转换编码的工具 372 大量转换编码 373

Chapter 18 主机系统管理 18.1 账号管理 376 18.2 DNS自动产生器 389 18.3 DHCP自动产生器 402 18.4 检测主机存在 407

Chapter 19 Bash在TCP/IP方面的运用 19.1 Bash网络转向 416 wget.sh程序说明 418 Echo服务器 419 19.2 自动登录FTP 421 19.3 自动登录SSH 424

Chapter 20 主机安全管理 20.1 收集主机信息 428 列出主机IP 428 找出主机名 428 列出核心信息 430 列出内存使用情况 430 列出磁盘种类 432 列出磁盘容量 433 列出磁盘用量 435 列出主机系统平均负载 437 列出系统中的隐藏文件 / 有特殊权限的文件 438 列出占用磁盘空间较多的目录 440

20.2 主机服务信道 441 基本概念 441 通信状态 443 管理主机服务信道 444 20.3 限制连接来源 446 使用TCP Wrapper 446 自动建立hosts.allow和hosts.deny 449 20.4 限制连接来源：使用iptables 452 关于Netfilter 452 关于iptables 452 iptables的语法 453 iptables的用例 454 自动建立iptables配置文件 455 20.5 文件完整性审核 465 关于checksum 465 使用md5sum 466 使用sha1sum 466 使用Shell自制文件完整性的审核工具 467 20.6 邮件通知 470 邮件方法 470 监测分区磁盘空间使用量 471 安排邮件内容的格式 475 使用Bash Shell邮件和附件 478 二进制文件编码 / 译码的方法 479 邮寄附件 480

Chapter 21 备份 21.1 使用tar做备份 488 基本用法 489 使用tar做完整备份 492 完整备份和差异性备份 493 使用tar进行差异性备份 494 还原备份文件的方法 494 21.2 使用磁带备份 495 操作磁带的方法 495 建立常用函数 501 21.3 使用cpio做备份 502 关于cpio 502 基本操作法 503 使用cpio备份到磁带 504 使用cpio备份到远程的主机 504 21.4 使用dump/restore做备份 / 恢复 505 关于dump和restore 505 /etc/fstab和dump的关系 505 dump和文件属性 505 dump的基本操作 506 备份至磁带 509 21.5 使用cp、dd做备份 509 使用cp 509 使用dd 510 21.6 使用光盘做备份 513 21.7 使用rsync做备份 514 关于rsync 514 设定rsync server 515 设定rsync client 517 安全性 519 rsync设定产生器 520 附录 Bash 4新功能

《实战Linux Shell编程与服务器》

编辑推荐

架构、流程、管理 系统维护 备份还原 安全管理 主机管理的上乘之道，就在于尽量让电脑去完成所有的琐事。要达到这个境界，管理者必须具备相当程度的Shell操控能力与Shell程序设计能力，卧龙小三拥有丰富的主机管理实务经验，以及多年的教学经验，相信这本《实战Linux Shell编程与服务器管理》必能引领读者轻松跨越Bash Shell程序设计的门槛，摆脱菜鸟劳碌的命运，达到轻松管理，主机不出包的境界。

《实战Linux Shell编程与服务器管理》特点： 完整涵盖Bash 3.x的各项功能，并介绍最新Bash 4.0的重要功能。 由简入繁、循序渐进，建立扎实的Bash Shell程序设计基础。 各章提供许多范例，充分展示Bash Shell程序设计的技巧。 带领读者学习如何设计自动化程序，轻松解决问题，增进工作效率。 还包含许多管理实务的技巧，可快速提升读者主机管理的能力。

精彩短评

- 1、 实战shell脚本
- 2、 模糊的很，真的初学者会被弄的一塌糊涂。
- 3、 实战Linux Shell编程
- 4、 买给我哥的
- 5、 改天要再看遍。。
- 6、 当初是把书中的命令都敲了一遍
- 7、 还行，前面讲得比较基础，后面就开始习惯性地贴代码了！
- 8、 可能是因为自己前面看了《linux命令行和shell脚本编程宝典》，感觉浪费钱了。。。。
- 9、 实用性比较强。
- 10、 我的shell启蒙书籍。致敬。
- 11、 很不错的一本数
- 12、 刚入手还没细看
- 13、 内容枯燥无味，组织混乱之极。

这个卧龙小三肯定不是鸟哥的私房菜里面提到的那个人！

- 14、 市面上少数写的通俗易懂的关于Shell编程的书。而且内容也比较实用。
- 15、 不错不错啊，确实好书，发现湾湾在linux这块还是书不少啊。
- 16、 内容不是很充实，字体好大，里面排版也不让人满意，对小三的书感到失望，认识小三是因为鸟哥，在这里还是要感谢鸟哥让我们入门linux
- 17、 我大学时候读过的一本书，内容很翔实周全，对于那时候初涉Ubuntu的我来说非常有帮助。
- 18、 鸟哥的私房菜基础篇讲Shell不是很清晰，这本书讲得比较实用，示例脚本比较多，虽然是2010年出的书不过内容值得一看
- 19、 dsdd
- 20、 读了一点，感觉没意思呀
- 21、 linux shell 编写必备入门.....
- 22、 很实用
- 23、 技术性数据，要好好看才能弥补不足

《实战Linux Shell编程与服务器》

精彩书评

1、内容枯燥无味，组织混乱之极。前面命令的罗列，环境变量的罗列，后面代码的罗列，这一切都深深地出卖了你——这个卧龙小三肯定不是鸟哥的私房菜里面提到的那个人！好吧，评论太短了作者分不清\$@和\$*的区别，认为没有引号时两者是有区别的这次评论应该够了

章节试读

1、《实战Linux Shell编程与服务器管理》的笔记-第56页

1) 内置命令

不用搜索\$PATH的命令 (*代表还未理解的命令) (查看所有的内置命令输入help)

alias 别名 / unalias 去除别名

bg 把作业放到后台 / fg 把后台作业放到前台

*bind 显示当前关键字与函数的绑定情况, 或将关键字与readline函数或宏进行绑定。

break 从最内层循环跳出

*builtin 运行一个内置Shell命令, 并传送参数, 返回退出状态0。当一个函数与一个内置命令同名时, 该命令将很有用

case

cd

*command 即使有同名函数, 仍然执行该Unix命令。也就是说, 跳过函数查找。

*compgen 结合Bash内置命令complete和compgen即可实现自定义Tab补全

complete

continue

declare 显示所有变量, 或用可选属性声明变量。

dirs 显示当前记录的目录 (pushd的结果)

disown 从作业表中删除一个活动作业。

echo 显示args并换行

enable 启用或禁用Shell内置的Unix命令

eval eval [args]: 把args读入Shell, 并执行产生的Unix命令。

exec exec command: 运行Unix命令, 替换掉当前Shell。

exit exit [n]: 以状态n退出Shell。

export export [var]: 使变量可被子Shell识别。

fc 历史的修改Unix命令, 用于编辑历史Unix命令。

for

getopts 解析并处理Unix命令行选项。

hash 控制用于加速Unix命令查找的内部哈希表

help

history

if

jobs 显示放到后台的作业

kill

let 用来计算算术表达式的值, 并把算术运算的结果赋给变量

*local 用在函数中, 把变量的作用域限制在函数内部。

logout 退出登录的Shell。

popd 从目录栈中删除项

pushd 向目录栈中增加项

printf

pwd

read 从标准输入读取一行, 保存到变量var中

readonly 将变量var设为只读, 不允许重置该变量

return 从函数中退出, n是指定给returnUnix命令的退出状态值

set/unset 设置选项和位置参量/取消指定变量的值或函数的定义

shift

shopt
source
suspend 终止当前Shell的运行（对登录Shell无效）
test 检查文件类型，并计算条件表达式。
time /times 打印设置命令执行的real user sys时间/显示由当前Shell启动的进程运行所累计用户时间和系统时间。（书中没讲times）
trap trap [arg] [n]：当Shell收到信号n（n为0、1、2或15）时，执行arg。
type 显示Unix命令的类型，例如：pwd是Shell的一个内置Unix命令。

```
-----  
ego@J:~$ type times  
times 是 shell 内嵌  
ego@J:~$ type time  
time 是 shell 关键字  
-----
```

typeset 同declare。设置变量并赋予其属性
ulimit 显示或设置进程可用资源的最大限额。
umask
until
wait 等待pid号为n的后台进程结束，并报告它的结束状态
while

2) 命令行程序

cp date who w ls cat

date
date + '时间格式'
date + '%d%H%M' 日 时 分

wc
wc -l file 显示file文件内容有几行
wc -c file 显示file文件有多少个字符
wc -w file 显示file文件有多少个单词。

find
很复杂的命令

tar
普通命令行 tar xvf test.tar.gz (-C /tmp) 解压（到/tmp）
tar zcvf test.tar.gz /tmp/123 将/tmp/123里的内容压缩到test.tar.gz的包里

另一种方式 tar cvf - /tmp/123 | gzip -9 > /tmp/save.tar.gz
将/tmp/123打包，管道到gzip，使用最大的压缩比9，压缩到save.tar.gz。此种方式不支持选

项z

gzip -9dc save.tar.gz | tar xvf -
同样不支持选项z，用gzip -9dc解压，-d表示解压，-c表示解压至标准输出，接着通过管道，交给tar包。-表示由标准输入读取数据

```
-----  
一个简易的备份脚本
```

```
#!/bin/bash
A=/tmp/123
B=/tmp/456

cd $A
tar cvf - . | tar xvf - -C $B
```

脚本有不少缺陷，但是基于这个脚本，慢慢扩增，即可成为一个全备脚本

basename 取得路径文件名中最后的文件名部分
dirname 取得路径名称中的目录部分

sort 排序
sort file 升序排序（按照ASCII字符，做大小比较）
sort -r file 降序排列
sort -n 升序排列 以字符串的数值大小为比较的标准
sort -k 3 file

<http://www.linuxfly.org/read.php?186> 讲的很详细

uniq 去除重复行
uniq file 去除重复行
sort file | uniq 排序file，并消除重复行
sort file | uniq -d 排序file，只显示重复行
sort file | uniq -c 排序file，消除重复行，并且显示重复的次数

cut 抽取文件部分
cut -c2 file 抽取file的每行第二个字符
cut -c2-10 file 抽取file的每行第2-10个字符
cut -c9- file 抽取file的每行的第九个之后的字符（包含第九个）
cut -c1-3,22- file 抽取file的每行1-3，22之后的字符
cut -d: -f1 passwd 抽取passwd每行的第一个字段
-d: 表示cut用：作为分割符，-f1是第一个字段的意思。
cut -d: -f2,4 passwd

paste 对文件以行和行的方式合并
paste file1 file2 file1和file2每行合并，默认以tab分割
paste -d '#' file1 file2 file1和file2每行合并，以#分割
paste -s file1 file1的每行合并到一行，默认以tab分割

tr 转换或删除字符
tr r R < /etc/passwd 把passwd文件中r换成R
tr ':' '\n' file 把file里的，替换成\n
tr -d k 删掉所有k字符
cut -d: -f1-6 /etc/passwd | tr: '#'（注意空格啊！！！！）取出passwd的1到6列，并改以#分割。
tr '[a-z]' '[A-Z]' < /etc/passwd 将所有的小写变成大写

`tr -s "" < file` 将多余的空白删除，只有一个空白。（未理解）

`grep` 过滤，看起来简单，其实用起来很复杂

`grep A *` 当前文件夹里，将含有A的文件和行显示出来

`grep -i A *` 将含有A和a的文件和行显示出来。-i意思是不区分大小写

2、《实战Linux Shell编程与服务器管理》的笔记-第31页

1) 通配符

* 代表任意字符，可以为空字符

? 代表一个字符，不可以为空字符

2) 转义字符,续航字符

\

3) 字符合集

字符合集就是[]中所列的字符范围中的某一个字符，长度为一，- 表示字符范围
[xyz] x 或 y 或 z

[a-z] 英文小写

[A-Z] 英文大写

[a-zA-Z] 英文大小写字母

[0-9] 数字

[a-zA-Z0-9] 英文大小写数字

字符合集中的“非”

[!0-9] 不是数字

[!a-z] 不是小写字母

... 其他类推

当需要匹配 - 这个符号的时候，将 - 放在最后

[a-z_-] 小写字母，_，-

当 ! 不在字符合集的开头，或者被 \ 转义的时候，就代表一个符号，而不是非的意思

[0-9!] 数字，!

[!\0-9]

当 * 在字符合集里时不代表匹配的意思，就是一个符号的意思

[123.*] 就是 1 2 3 . 和 *

4) 括号扩展

```
echo {a,b,c}_test ---&gt; a_test b_test c_test  
echo {1,2,3,4,5,6,7,8,9}\{1,2,3,4,5,6,7,8,9} ----&gt; 九九乘法表
```

注意：书35页中端部分说 此处*需要转义 似乎是不准确的

bash 4.1(ubuntu 10.10), bash 3.2(RHEL5) 中 echo a*b 和 echo a\b是完全一样的。
但是的确echo *的结果和ls是同样的。

5) < > 连用

格式：命令 < 文件 > 转入文件

```
ego@J:/tmp/123$ sort &lt; testfile  
04test1  
11test1  
28test1  
35test1  
45test1  
46test1
```

sort < testfile > file file文件里就被写入上面sort排序的结果。

6) 管道

命令1的输出结果，通过管道，变成命令2的输入

cat testfile | sort > file3 ----> 过程是读取testfile，sort排序，写入file3文件。

管道的组合使用，经典样例！！！！需要牢记！！！！

```
grep '".*" 4[0-9][0-9]' access_log | grep -o '".*" 4[0-9][0-9]' | sort | uniq -c | sort -n | tee /tmp/log.txt
```

access_log是apache日志文件，将这行命令分解如下

```
grep '".*" 4[0-9][0-9] access_log 找出含有4xx错误信息的内容  
grep -o '".*" 4[0-9][0-9]' 将前面的输出结果过滤，取出符合样式的字符串。  
sort 排序  
uniq -c 去除重复行，并且统计重复次数  
sort -n 再交给sort，按照重复次数排序  
tee /tmp/log.txt 最后交给tee写入文件，并且显示。
```

7) 前台工作，后台工作

&fj bj 具体google，查man

3、《实战Linux Shell编程与服务器管理》的笔记-第43页

1) Bash结构：变量设定，内置命令，语法结构，函数，其他命令行

2) 执行方式：`./test.sh`

`. test.sh` (与前一种其实是有区别的，不会fork出子shell)

`source test.sh` (同样不会fork出子shell)

`/root/test.sh` (需要执行权限的)

`bash/sh test.sh`

3) 排错方法

`bash -v`

`bash -n`

`bash -x`

4、《实战Linux Shell编程与服务器管理》的笔记-第16页

(此方法和书中不同)

FreeBSD默认是用的csh

1) 安装bash

`cd /usr/ports/shell/bash`

`make install clean`

2) 修改配置文件

装好以后，使用`chsh -s /usr/local/bin/bash`就可以把默认的shell改成bash了。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com