

第3波

计算机技术入门提高精通系列丛书

UNIX初学者

实用手册

杨景翔
孟朝蓬
王冰峰

编著
改编
审校

人民邮电出版社



6.81
/1

TP316.81
YJX/1

计算机技术入门提高精通系列丛书

UNIX 初学者实用手册

杨景翔	编著
孟朝蓬	改编
王冰峰	审校

人民邮电出版社

计算机技术入门提高精通系列丛书

UNIX 初学者实用手册

- ◆ 编著 杨景翔
 改编 孟朝蓬
 审校 王冰峰
 责任编辑 顾 翀
 - ◆ 人民邮电出版社出版发行 北京崇文区夕照寺街 14 号
 北京鸿佳印刷厂印刷
 新华书店总店北京发行所经销
 - ◆ 开本:787×1092 1/16
 印张:12
 字数:294 千字
 印数:1-5 000 册
 1997 年 8 月第 1 版
 1997 年 8 月北京第 1 次印刷
 著作权合同登记图字:01-96-1206 号
 ISBN 7-115-06551-9/TP·457
-

定价:19.00 元

内 容 提 要

随着计算机网络应用的日益广泛,越来越多的人需要了解有关 UNIX 系统的知识。UNIX 操作系统是一种通用的交互式分时操作系统,被广泛应用于小型机、工作站和计算机网络。由于 UNIX 系统比较复杂,使用起来难度较大,令不少初学者摸不着头脑。本书是一本出色的介绍 UNIX 系统使用的入门读物,全书通过生动活泼的语言,众多简单实用的范例以及整齐有序的版式设计,使读者可以轻松愉快地学习各种 UNIX 的知识,透过其严肃、复杂的命令外表,快速掌握 UNIX 的操作技巧。

本书适合 UNIX 初学者阅读,对于已经使用过 UNIX 系统的用户来说,也能够使读者进一步丰富更多的专业知识。

版 权 声 明

本书繁体字版名为《UNIX 初学者使用手册》，由第三波文化事业股份有限公司出版，版权归第三波文化事业股份有限公司所有。本书简体字中文版由第三波文化事业股份有限公司授权人民邮电出版社出版。专有出版权属人民邮电出版社所有，未经本书原版出版者和本书出版者书面许可，任何单位和个人均不得以任何形式或手段复制或传播本书的一部分或全部。

版权所有，侵权必究。

出版说明

在计算机技术飞速发展的今天,为了进一步向全社会普及计算机知识,提高计算机应用人员的技术水平,使计算机在各个领域发挥更大作用,也为了促进海峡两岸计算机技术图书的交流,台湾第三波文化事业股份有限公司授权我社陆续组织出版该公司的部分计算机技术书籍。

在组织出版过程中,我们请有关专家在尊重原著的前提下,进行了改编,并对有关图文进行了核对和精心制作。

由于海峡两岸在计算机技术名词的称谓上差异较大,改编者依照有关规定和我们的习惯用法进行了统一整理。

对原书文字叙述中由于海峡两岸不同的语言习惯而造成的差异,我们的处理原则是只要不会造成读者理解上的歧义,一般没做改动,以尊重原著写作风格。另外改编时对原书的一些差错及疏漏之处做了订正。

由于本书改编和出版时间紧张,如有差错和疏漏,敬请读者指正。

人民邮电出版社

1997.4

●第一章 准备工作	1
1.1 登录(Login)	1
1.2 文件系统(File System)概述	10
1.3 退出(Logout)	15
●第二章 常用的 UNIX 命令	17
2.1 随便先看看	18
2.2 不必急着学会	20
2.3 文件权限	36
2.4 再接再厉	38
2.5 魔数	57
2.6 快学完了	60
2.7 1752 年 9 月为什么少 11 天	63
●第三章 UNIX 简史	65
3.1 失败的计划	65
3.2 诞生	66
3.3 欣欣向荣	67
3.4 Richard M. Stallman	68
3.5 编辑器	72
3.6 你可能不知道	73
●第四章 你的壳 C Shell	75
4.1 一个简单的.cshrc 例子	75
4.2 预定义变量(Predefined Variables)	77
4.3 别名(alias)与历史(history)	83
4.3.1 别名(Alias)	83
4.3.2 历史(History)	84
4.3.3 健步如飞	84
4.4 环境变量(Environment Variables)	97

●第五章 看看几个实际的例子.....	103
5.1 第一个例子——.cshrc	103
5.2 第二个例子——.login	111
5.3 第三个例子——.logout	116
5.4 有些事情你最好先知道	116
●第六章 UNIX 工具的浏览	123
6.1 V - eye	123
6.1.1 小心你的键盘	124
6.1.2 vi 的模式有两种	124
6.1.3 真的来玩玩	124
6.1.4 编辑模式下的 vi	130
6.2 sed 和 Regular Expression	132
6.2.1 寻找与替换	132
6.2.2 匹配行首: ^	133
6.2.3 匹配行尾: \$	134
6.2.4 匹配任一字符:	134
6.2.5 多种选择的匹配法: [...]	135
6.2.6 匹配零或多个字符: *	136
6.2.7 非常奇怪的表示法	139
6.3 关于 sed 的其他用法	144
6.4 grep 命令简介	146
6.5 cut 的用法	149
6.6 顺便谈谈 paste	152
6.7 tr 命令简介	154
●第七章 学写 C Shell Script	159
7.1 你的第一个 Shell Script	159
7.2 用 Shell Script 更改文件名	165
●第八章 X Window System	169
8.1 打开窗口(Open the Window)	170
8.2 X Resources	172
8.3 更多一点知识	175
●第九章 再介绍些资料.....	177
9.1 介绍一些有名的程序	177
9.2 参考信息	181

第一章

准备工作

要使用 UNIX, 你得先要有个帐户 (account)。你必须向你的系统管理员申请, 他会给你一个“用户名”(username) 以及一个“密码”(password)。密码用来保护你的文件, 保护系统, 保护一切。不要忘了, UNIX 是多人使用的系统, 除了你, 还有许多人也用同一个系统。为了区别身份, 每个使用者都会有个用户名, 就像每个人都有个名字一样; 密码就更不用提了, 想想你的金融提款卡就知道了!

1.1 登录(Login)

有了帐户(account), 上机的头一件大事就是登录(login)。这一节将介绍与登录有关的步骤及相关文件。你将了解 UNIX 如何识别你的身份, 如何把你安置在系统中, 如何给你一个“家”! 特别先提醒你, UNIX 区分大小写, 而且几乎所有的命令、程序都是小写的! 这一点跟 DOS 有很大的不同。另外, UNIX 里的文件名称没有什么限制, 而且文件名长度可以到 256 个字符。有些老的 UNIX 只能接受最多 14 个字符的文件名称, 但这种系统已不多见。你爱取什么名字就叫什么名字, 不必有什么讨厌的限制(在 Windows 95 以前的 MS-DOS 里面, 文件名只能有 8 个字符, 扩展名最多只能有 3 个字符)。

LOGIN

登录(login)后,系统会显示某个(段)符号,例如:

%

或者

ccsun2 /1 >

这个符号叫“系统提示符”(System Prompt)。此时你已经在 UNIX 里面了。更精确地说,你已经是在一个 Shell 里面了。Shell 就是“壳”,它担任一个翻译者的角色,你给的每个命令都由 Shell 负责将它解读并做处理。

Prompt

到这里为止,好像跟 PC 的 DOS 差不多? 我们使用 PC 的 DOS 时也是在“>”后面下命令。对! 你可以这样说,但 Shell 本身却也是 UNIX 众多“命令”(应该说是“程序”)中的一个。所以,你可以使用不同的壳(Shell)来担任你的翻译器。你可以使用 /bin/csh(C Shell),也可以用/bin/sh(Bourne Shell),或者由网络上取得的 tcsh、bash 等等,甚至你也可以自己写一个 Shell。这一点和 DOS 就不同了,在 UNIX 里,你可以选择你的翻译器(Shell),而在你申请帐号后,你的系统管理员会先替你指定一个缺省的 Shell。一般来说,如果你用的是 Sun 工作站,那会是 C Shell。总之,只要你登录进了系统,你就有个“壳”。

SHELL

那么,你如何知道你用的是哪一个 Shell 呢? 你可以看一看/etc/passwd (用“more /etc/passwd”命令)这个文件,它的内容如下:

.....

.....

john:ACrBqjTb8JmxQ:149:200:John Cooley:/home/users/john:/bin/csh

cpy:,xcrtgtKWJTUA:172:200:C. P. Y:/home/users/cpy:/bin/tcsh

math:po.isgMVOPEw:110:200:Guess Who?:/home/users/math:/bin/sh

smith:ouzoilEiDuJuk:129:200:Bob Smith:/home/users/smith:/bin/csh

student1:ou..41EiDuJuk:159:300:TalkingHead:/home/users/student1:/bin/csh

.....

.....

这个文件每一行定义一个用户,以“:”为分隔符号,定义一个用户的 login name(用户名)、密码、登录后的“家目录”(Home directory)以及登录后要率先执行的“东西”等等。由左至右以“:”分隔的七项,分别说明如下:

- ① 用户名(username):第一项是使用者登录时的用户名,一般来说,使用者只能接受系统管理员给的名字,不能随意任你选。比如在学校,大都会用学号来命名,没什么大道理,就是方便管理而已。除非你有什么后门(跟系统管理员是好朋友啦,什么的……),否则,这个项目是无法改变的。如果你觉得你的用户名很拙,很差,实在不喜欢,对不起,你还是得忍受它。把你心目中想要的好名字用在下面说的第5项,全名(full name),比较实际。
- ② 密码(password):第2项是使用者的密码,经过特殊编码之后的文字。基本上你可以假设没有人可以从这些乱七八糟的字符还原成你在登录时所输入的密码。密码是可以由你自己随意更改的,但要切记,不要用太简单的密码,像直接使用用户名当作密码,或者用简单的数字、英文姓名,像 1234567、Annie、James,都不是好的密码。密码尽量复杂,可以是 a5g2s7ts、s \$ bb% d! &。当然,前提是你自己要记得住,在你下次登录时,都不要忘记。系统管理员最讨厌的事之一就是三天两头有使用者跑来跟他说“密码忘记了”。
- ③ 用户识别符(user id):第3项是使用者的“身份证号码”,就像用户名(username)是你的姓名一样,这个号码也是唯一的。换句话说,在你所使用的系统中,没有其他人跟你有一样的名字,也没有使用者的“用户识别符”(user id)跟你一样。而事实上,在 UNIX 里面,系统真正用来区别身份的是这个用户识别符。你可以用“id”命令来看你的用户识别符(user id)和下面说的组织别符(group id)。

id

- ④ 组织别符(group id):第4项是使用者所属的群体编号,就像每个学生都会隶属于某个班级一样。在 UNIX 里,每个使用者也会被分类到他所属的组(group),而每个这样的组都有个独一无二的编号,就是 group id。这种把使用者再分类的作法是在让使用者可以更方便管理他们的文件,比如说,同属一个组的人可以共用某些文件,并拒绝其他组的人读取这些文件。每个使用者一定属于至少一个组,换句话说,一个使用者可能同时属于好几个组。就像一个学生可以是一年级1班的学生,也同时是其他协会会员

一样。你可以用“groups”这个命令来看你是属于哪一个组的。这个命令会显示出你所在的组,可能一个,两个,也可能没有。为什么没有呢?那是因为系统管理员没有帮你分类的缘故^①。你可以看一下 /etc/group 这个文件,它的样子有点像现在讲的 passwd 文件,但只有 4 栏,依次为:“组名称:密码:组编号:该组成员 1,成员 2,...”。这个文件也是系统管理员设定的,所以说,你自己并不能决定你要加入或退出某个组。

Group

⑤ 用户全名(full name):第 5 项是使用者可以自己更改的项目,一般都是用使用者自己的姓名,以方便知道用户名(username)与真人之间的关系。另外是在你寄出电子邮件时,收信人可以知道寄信的人是谁,而不必从用户名去猜测。当然,你可以随意替自己取个响亮的名字,像 Albert Einstein 啦,Bill Clinton 啦,都可以。

⑥ 家目录(home directory):第 6 项是定义你的所谓“家目录”。这好比你家的户口簿上除了要登录你的身份证号码外,还要详细记录你家地址一样。这个家目录就像你被分配到的公寓一样,只要你一登录到 UNIX 里,系统就自动把你“安置”到这个属于你的地盘。这也是不能随你更改的,由系统管理员分配。

HOME

这个“家”的概念,在 UNIX 里很重要,因为 UNIX 系统是一个所谓多人多工的系统。也就是说,同一个系统内可能有上百人同时在使用,而且,每个人都享有同样的系统资源。为了避免不必要的麻烦,每个人都会被分到一块磁盘空间当作“家”,每个人只能在自己的“家”里面为所欲为,不可以干扰到别人。所以你的一切文件,包括垃圾,都只能放在自己家里。你有绝对的权力来管理你的家,把门关上或把门打开,格局也随你。当然如果你够慷慨,你也可以打开你的家门让别人进来胡搅。但一般情况下,个人的家目录(home directory)别人只能跳进去看看文件,而不能做什

^① 没帮你分类,那么你在/etc/group 这个文件中就看不到你的名字。但是你还是会有个组织别符(group id),系统就用这个号码来当你的组名字。当你用“groups”这个命令时,就看到数字。

么事情。有的人基于某些理由,更是连家门都上锁了,别人连瞄一眼都不行。

⑦ 壳(shell):

第7项是定义该使用者在登录完成之后,所要执行的东西,一般是定义一个使用者所用的壳(Shell)名称,也就是指定命令的翻译器。这一项你可以自己指定,在你申请帐号时系统管理员给的只是暂定的,换不换随你。

以上介绍了 /etc/passwd 这个文件,其中有3项可以由使用者自己选择,即 密码(password)、用户全名(full name)、壳(shell)。

密码的更改是你一定要会的,而且是经常要做的。有的系统甚至强迫使用者每隔一段时间就要换一次密码。更改密码用下面这个命令:

```
% ypasswd
```

```
Changing NIS password for Jacky on roxy.
```

```
Old password:
```

```
New password:
```

```
Retype new password:
```

```
NIS entry changed on roxy
```

如果无效,则用以下命令:

```
% passwd
```

(为什么有两个改密码的命令呢?稍后再作说明。)

它会要求你先输入旧密码,再连续输入两次新密码(第二次是用来确认),以完成更改的动作。

值得一提的是,你的密码虽然可以很长,但是绝大多数的 UNIX 系统只认得前8个字符,所以,“MtosaJKopq”与“MtosaJKo12”是等效的,超过8个字符的密码只是为了好记或练打字罢了。另外有些系统规定密码至少要有6个字符,或规定密码之中必需要含一个以上的数字且含2个以上的英文字母等等,这些规定都是用来减少密码被破解的机会。由此也可以看出 UNIX 系统很怕密码被破解的事,尤其系统管理员的密码,更是要小心。

用户全名(full name)的更改使用以下命令:

```
% chfn
```

它会先显示你现在的用户全名,然后要求你输入新的全名。至于用户全名可不可以用中文,要视系统而定,有些系统会因中文码而产生不可预期的结果。在确定这件事之前,最

好不要尝试,先问问你的系统管理员再说。

壳(shell)的更改使用以下命令:

```
% chsh
```

它也会先显示目前你用的壳,如 `/bin/csh`,然后要求你输入新的壳名。要注意的是,这里的壳名称必须是一个系统认可的名字,例如 `/bin/csh`,或 `/bin/sh` 等等。

至于哪些是系统认可的壳名字,就要看 `/etc/shells` 这个文件了。如果没有这个文件,那系统只能接受 `/bin/sh` 及 `/bin/csh` 当作你的缺省壳(default Shell)。这就是说,你就没法将 `/etc/passwd` 里面,你的那一行最后一项改成 `/bin/sh` 或 `/bin/csh` 以外的东西。这并不表示你就没办法用其他壳,例如你自己做了一个壳或者在网络上取得其他壳(你的系统管理员当然没法知道世界上又多了哪种壳),而且要把它放到 `/etc/shells` 这个文件中去。所以,要求系统管理员随时按使用者的要求去更改 `/etc/shells`,事实上是不切实际的。这样说来,刚刚说使用者可以自由选择自己的壳是不是就有点骗人了?当然不是,在这里说的是用“chsh”命令来改变使用者的缺省壳(default Shell),这个缺省壳是使用者刚进 UNIX 时就用到的。进了这个缺省壳后,你还可以再进其他你自己喜欢的壳,并不受限制。所以即使你更改不了缺省壳,那也不是什么大不了的事,顶多在登录之后再进你想用的“壳”就可以了。至于在无法改变缺省壳的情况下,怎么进(改用)你想要用的壳,后面再谈。另外要提的是,如果你的机器上 `/etc/passwd` 这个文件很小,只有短短几行,而且也没看到你的用户名在上面,那么再试试用“ypcat passwd”命令看看,原因也是稍后再说。

逐项说明完 `/etc/passwd` 这个文件中各项的含义,我们再回头看看前面的例子。假设你是“student1”那个用户,在该行的第一栏即是你的用户名,第2栏是编码过的你的密码,第3栏是用户识别符(user id),第4栏是组织别符(group id),第5栏是你的全名(full name),第6栏是你的“家”目录,最后一栏是你一登录后,系统要执行的东西。这里,最后一栏是设定为 `/bin/csh`,所以,你登录后就直接进到 C Shell 了。反之,如果在 `/etc/passwd` 中该行的最后一栏设定是 `/bin/sh`,那么,你进 UNIX 系统后,就会直接进到 Bourne Shell 内。`/etc/passwd` 这个文件只有系统管理员(System Administrator, System Manager, root)可以直接更改。但前面也提到,你仍然可以用下表中的一些命令来更动部分原有的设定:

项 目	用来更改原来设定的命令
密码(password)	“yppasswd”,如果没有这个命令,则用“passwd”
全名(full name)	“chfn”
壳(Shell)	“chsh”

借助这些命令,你可以改变 `/etc/passwd` 中自己那一行的3个栏目。其他栏目的更改,只有系统管理员才能做,一般使用者无法更改。

好了,现在来说明一下,为什么会有两个改密码的命令(一个是“passwd”,另一个是“yp-passwd”)。passwd 前面加个“yp”是“Yellow Pages”的缩写,它是 Sun 公司所发明的一种系统管理方式。后来因为“Yellow Pages”是英国某个单位的注册商标,就改名叫做“NIS”,是“Network Information Service”的缩写。

NIS

简单的说,你现在可以在很多台机器用同一个用户名、同一个密码来登录进入系统,改密码时也只要在其中一台机器更改,不必每台机器都做同一个操作。而系统管理员也只需在一台机器上建立每位用户的资料,如 /etc/passwd 这个文件,不必每一台机器都去做同样事。可以让使用者在随便一台机器登录进系统,不必受限只能由某特定机器登录,这些都是 NIS 的功能。由于这个发明,系统的运行更有效率,管理上也更容易,所以现今的很多 UNIX 系统都采用它。也因此,原来在 NIS 发明前的改密码的操作就有些不一样了。在很早以前,每一台 UNIX 机器都有自己的使用者,也就是说,每一台机器都有个 /etc/passwd 文件。如果你要使用机器 A,那你得在机器 A 的 /etc/passwd 文件上有名字;要使用机器 B,得在机器 B 的 /etc/passwd 文件上有名字。要改密码时,在机器 A 上用:“passwd”,改了机器 A 的密码,机器 B 的密码还没变动呢!

现在有了 NIS,用户的资料都集中放在所谓的 NIS 服务器(server)上面,其他机器只是所谓的客户(client),没有必要保留像 /etc/passwd 这样的文件。这些资料只要在必要时由 NIS 服务器提供即可,所以你不一定可以在你所使用的机器上找到真正的 /etc/passwd 文件,也就是说你看到的 /etc/passwd 文件只是该机器自己使用的而已。而这台机器可能不是 NIS 服务器,所以你可能看到一个很简单的 /etc/passwd 文件,那里面没有你的用户名,而你依然可以登录。

那么你要问,使用者怎么知道哪一台机器是 NIS 服务器,怎样看看真正的 /etc/passwd 文件是什么样子呢?如果你的系统运行 NIS 的话,可以用“ypcat passwd”来得到一个完整的 passwd 文件,或者用“ypcat passwd | grep 你的用户名”来得到你自己应该在 /etc/passwd 文件中的那一行。

而旧的 passwd 命令和新的 yppasswd 有何不同呢?简单的说,yppasswd 更改的是 NIS 的资料,而 passwd 改的是你用的那台机器的 /etc/passwd 资料。如果你用的这台机器上面没有你的资料,那表示你的资料应该在 NIS 服务器 那边,系统会自动去更改 NIS 资料。这时,passwd 和 yppasswd 的作用就一样了。反之,万一在你所用的那台机器上 /etc/passwd 有你的资料,那 passwd 命令就更改这个文件,而不会去更改 NIS 的资料,这时跟 yppasswd 的结果(修改 NIS)就不同了。

无论如何,一般如果是运行 NIS,还是用 yppasswd 这个命令来修改密码为好。而且,只

要在其中一台可以登录的机器修改就可以了。万一没有 `yppasswd` 这个命令或者没有 NIS, 再用 `passwd` 也不迟。

好啦, 上面这一段算是把登录的操作讲完了, 如果大家还想进一步了解 NIS 或是其他的细节, 可以找找其他相关书籍, 或是去询问你的系统管理员。本书以 C Shell 为说明对象, 如果你不是用 C Shell 或 TC Shell, 有部分内容可能会有些不对, 请读者自己注意。而在容易出错的地方, 笔者也尽可能先点出来。

回到前面的话题, 说到登录后的系统提示符: `prompt`。有人不喜欢既定提示符的样子, 想用个特别的, 例如改成:

```
I Hate UNIX >
```

这当然是可以的, 下面这样的命令就可以做到:

```
set prompt = 'I Hate UNIX > '
```

关于设定环境的其他事项, 在这里先略过不谈, 在后面 C Shell 那章再好好研究。

假设你的缺省壳(default Shell) 是 C Shell (`/bin/csh`), 那么, 从你登录开始你就脱离不了它。你下的所有命令都跟它有关。UNIX 迷人的地方之一便是, 她允许你用很简单的方法重新设定你自己的环境。

某甲下的命令“`del ABC`”可能是把 `ABC` 这个文件删掉, 但某乙下同样的命令“`del ABC`”可能是把 `ABC` 这个文件打印出来。总之, 你可以替你自己的情况订制一个属于自己的 UNIX 环境。在前面提到, Shell 居于你和 UNIX 之间, 充当翻译员的角色。所以, 要替自己订制一个特别的环境, 你只要好好教育翻译员就可以了。像刚刚提到的例子, 某甲告诉他的翻译员说, “`del`”表示“删文件”, 所以当某甲下“`del ABC`”命令时, 他的翻译员(壳)就去执行“删文件”的动作。而某乙则是教育他的翻译员说, “`del`”是“打印文件”的意思, 所以, 当某乙下“`del ABC`”命令时, 做的却是“打印文件”的操作。所以, 你只要按你的喜好, 教育好你的壳, 以后你就可以轻轻松松指使它去做一些你想做的事, 使个眼色就好了, 而且这种教育工作只要做一次就好。除非你改变主意了, 要它换个样子, 否则, 这种教育工作不必再做。

这个教育工作很重要吧! 必定有个特别的文件来作为教案。对的, 这个教案就是“`.cshrc`”, 在你的“家目录”(Home directory) 里面。它用来设定你 C Shell 的相关环境。

另外值得一提的是, 在 UNIX 里面, 像“`.cshrc`”这种用来设定环境的文件几乎都是以“.”开头来命名的, 如“`.cshrc`”、“`.login`”、“`.xinitrc`”……等等。而这些以“.”开头来命名的文件是半隐含文件, 当你用“`ls`”来看文件名称时, 必须用“-a”这个选项(option)才看得到。

一般来说, 在你申请帐号时, 你的系统管理员(System Administrator, 以下简称 SA)都会帮你准备一个适用你所在计算机环境的基本“`.cshrc`”文件。

现在先不管“.cshrc”，因为在后面我们有专门介绍它的章节。我们先来把登录的操作讲完。可能有人已经发现，有些人一登录进系统就会自动执行一些程序，比如说，直接进到 X Window (或是 Openwindows) 里面，或者屏幕会出现个 “Good Morning” 的字样，而有的人却不会，为什么呢？这是另一个文件 “.login” 的功能。

顾名思义，“.login”就是在登录(login)时产生作用的。它紧接着“.cshrc”之后“执行”。为什么说“执行”呢？因为“.login”事实上是一个用户自己写的“程序”，一个以 C Shell 语言写成的程序(我们叫它 C Shell Script)。当你每一次登录时，“.login”就执行一次，而且只有一次，除非你用特别的命令再去改动它。

它跟“.cshrc”不同的是，“.cshrc”在每一次进 C Shell 时，都会被“看”一下，所以你每多进行一次 C Shell(比如说，在 OpenWin 里面又开了一个 comdtool、xterm 或者执行某个 C Shell Script)，“.cshrc”就多被“看”一次；而“.login”却不会。

有人大概还不清楚什么叫做“进入”Shell，进入一个壳？什么意思呢？现在假设你已经登录完，在 UNIX 里面了，你可以在系统提示符后面下这个命令：

```
% /bin/csh
```

然后是：

```
% exit
```

执行完后，你一定发现好像没什么事发生一样。是没什么事发生，只不过你又进了一个壳。这就好像一个经过多层包装的纸盒一样，你刚登录的时候是在最外层，然后如果你再执行如上的第一个命令(/bin/csh)，那就又往里头钻进一层，该层的环境跟在最外层是完全一样的，因而你不会感到任何差别，只不过在它外面，还包了一层壳；第 2 个命令是跳出所在的那一层，以这个例子来说，就是回到最外层。当然，你可以一直钻进去，就是重复执行命令“/bin/csh”，然后，你可以再用“exit”命令又回到最外层。不管你在哪一层，你都把它当作完全一样，因为它们都是一个壳，完全一样的壳！所以即使你忘了你钻进几层，不知道该“exit”几次，那也不要紧，留在那里也无妨。

前面不是说，改不了缺省壳(default Shell) 也没关系吗？缺省壳就是最外层的壳，你登录后会先在这层壳，然后，你可以再进你喜欢的壳。就呆在那里面不要出来，直到要离开 UNIX 为止，这样，就达到换壳的目的了！比如说，你在 /etc/passwd 的最后一项是被设定成 /bin/sh，那你又习惯用一个叫 tcsh 的壳，因为它有类似 DOSKEY 的功能，而偏偏你用“chsh”命令又没办法将 default Shell 从 /bin/sh 改成 /usr/local/bin/tcsh(假设 tcsh 是放在 /usr/local/bin 这个目录下)，因为你的系统管理员并没有把 /usr/local/bin/tcsh 放进 /etc/shells。没关系，以后 login 进来以后，就钻进 tcsh 里去，不要出来，一直到要注销(log out) 为止。怎么钻进去？用下面的命令就可做到：

```
% /usr/local/bin/tcsh
```