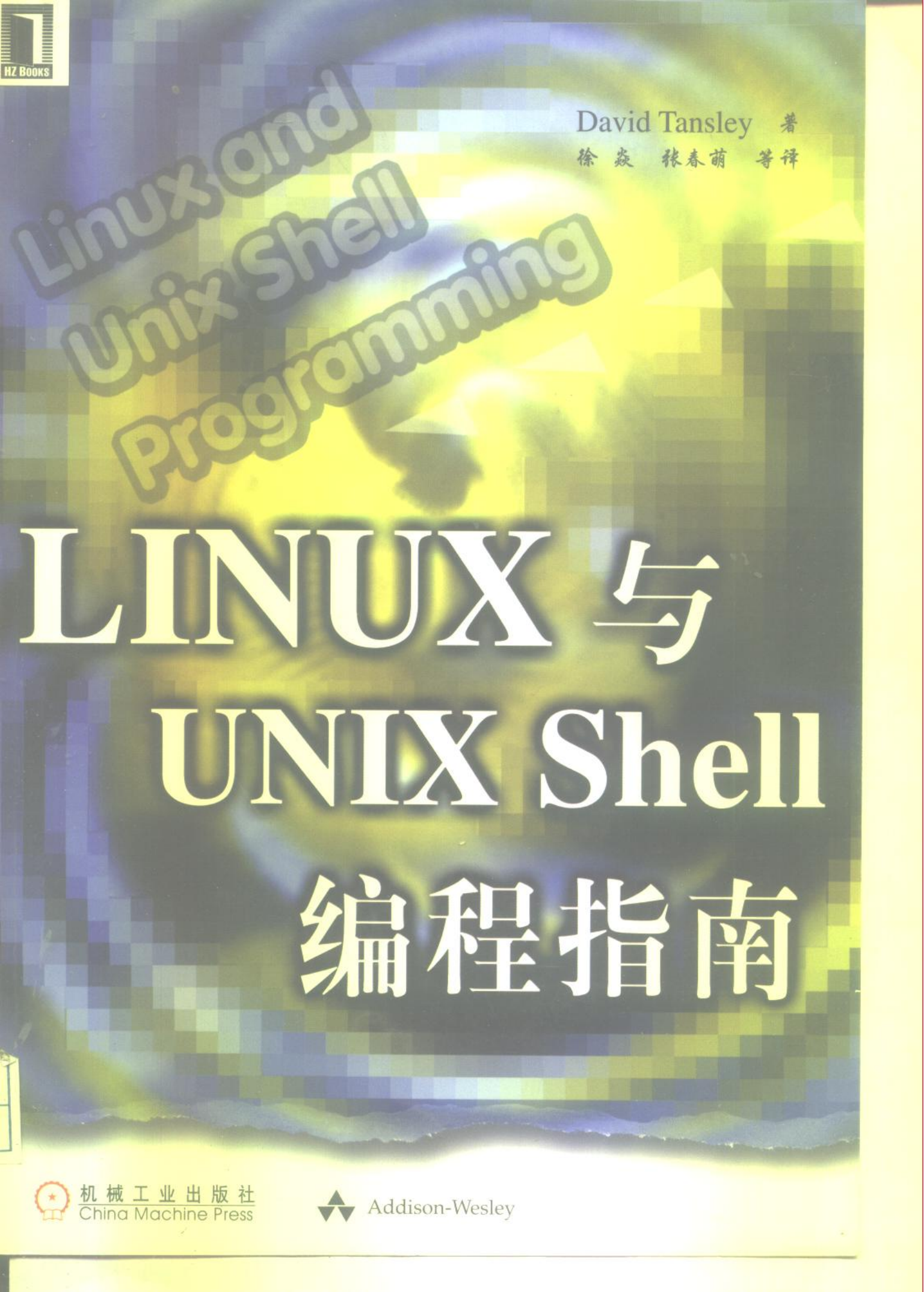
HZ BOOKS
David Tansley 著
徐焱 张春萌 等译
Linux and Unix Shell Programming
LINUX与UNIX Shell编程指南
机械工业出版社
China Machine Press
Addison-Wesley

LINUX与UNIX Shell编程指南

(美) David Tansley 著

徐 焱 张春萌 等译

机械工业出版社
China Machine Press

本书共分五部分，详细介绍了shell编程技巧，各种UNIX命令及语法，还涉及了UNIX下的文字处理以及少量的系统管理问题。本书内容全面、文字简洁流畅，适合Shell编程人员学习、参考。

本书版权登记号：图字：01-2000-0124

图书在版编目（ CIP）数据

LINUX 与UNIX Shell 编程指南/（美）坦思利（Tansley, D.）著；徐焱，张春萌等译. - 北京：机械工业出版社，2000.6

书名原文：Linux and Unix Shell Programming

ISBN 7-111-08041-6

1. L… II. ①坦… ②徐… III. UNIX操作系统 IV.TP316.81

中国版本图书馆CIP数据核字（2000）第23834号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码 100037）

责任编辑：赵红燕

北京昌平第二印刷厂印刷 · 新华书店北京发行所发行

2000年6月第1版第1次印刷

787mm × 1092 mm 1/16 · 23.5印张

印数：0 001-6 000册

定价：38.00元

译 者 序

shell是UNIX系统中重要的组成部分，用户通过它与系统内核交互，任何使用UNIX的用户都离不开shell。同时，UNIX shell又是一种独具特色的编程语言，它充分体现出UNIX简洁的特点，使用方便，功能强大，健壮性好。可以这样说，实现同样的功能，使用shell脚本几乎是最快捷、最简便的方法。因此，学习UNIX应当首先学习shell。

然而，很多UNIX用户对shell仍是一知半解，往往存在很多错误概念。本书可以帮助你澄清这些错误概念。本书主要涉及了Bourne shell，这是一种最“古老”的shell，同时也是兼容性最好的shell，它最初由AT&T的Stephen Bourne创立，后来出现的Korn shell、POSIX shell以及LINUX上的BASH shell都与之兼容。在理解Bourne shell后，就可以很容易地进一步学习其他shell。此外，本书还涉及了UNIX下的文字处理，以及少量的系统管理问题。

本书具有以下特色：

- 将UNIX shell中有关重要问题各自单独编排为独立的章节，便于读者学习和理解。
- 书中含有大量实用的例子。
- 书中充分体现了作者对UNIX shell的深入理解，读者还可以从中学习到很多系统管理经验。

的确，对于具有一定经验和富有耐心的UNIX用户而言，读在线手册可以获得最精确的定义。但是，读在线手册毕竟非常枯燥。你会发现本书更易于理解，书中所包含的经验和睿智也是无法从在线手册中获得的。

需要说明的是，UNIX系统及shell版本千差万别，在理解书中的精髓之后，很多具体问题还需要查阅相关平台的手册具体对待，不可完全照搬。

本书的第一、五部分的翻译主要由张春萌、刘作伟、付顺旗、阎济宙完成，第二、三、四部分的翻译由徐焱、肖仁春、白广齐、时向泉、魏立峰、朱敏完成。由于时间仓促和译者的水平有限，错误和疏漏之处难免，诚恳地希望读者朋友提出宝贵意见和建议。

2000年3月

前　言

这是一本关于shell编程的书，更准确地说，是关于shell和Bourne shell编程的书。

LINUX作为一种健壮而有生命力的操作系统，已经牢固地占据了当前的市场，shell编程也变得越发流行起来。尽管有越来越多的第三方供应商在销售他们的LINUX变体，LINUX仍然应当算是一种免费软件，没有人能够准确地说出目前到底有多少LINUX用户。尽管数年前就有一些IT顾问预言UNIX将让位于其他操作系统，但现在它依然像过去那样流行，并且还在发展壮大。

如果你想学习shell编程，本书正适合你；即使你已经入门，你仍然会发现本书既适合学习，也可以作为手边的参考书，因为书中有不少方便实用的系统管理小技巧和趣味脚本代码。

本书从一开始就基于这样6个主要目标:

- 使读者尽快掌握shell工具和shell编程。
- 使本书不但可以作为学习的工具，还可以作为参考书。
- 运用shell脚本使系统发挥出更加强大的功能。
- 展示清晰、易于理解的脚本。
- 尽可能地按照所涉及的主题和易于使用原则，每一章自成体系。
- 不但向读者提供shell脚本，还提供一些类似于运行级别脚本(rc.scripts)和cgi脚本的系统管理内容。

这类书籍往往有这样的恼人之处：为了省略区区几行代码，使得脚本的例子变得晦涩难懂。本书不存在类似的问题：书中所有的代码都简单有效。

看到这儿，你可能已经明白了为什么要学习shell编程；这很好。如果你不知道为什么要学习shell编程，让我来告诉你:

- shell是一种完整的编程语言；它含有循环，条件转移和判断结构，很容易使用。
- 可以快速地创建shell脚本。
- 可以应用shell脚本使烦人的手工作业自动化。

Bourne shell

Bourne shell是所有的UNIX系统都支持的标准shell，它也链接到了LINUX的BASH shell中。一本关于shell编程的书要想适合于所有业界领先的UNIX操作系统，就必须包含Bourne shell。当然，目前还存在其他几种shell，例如BASH shell、Korn shell、还有C shell。如果你熟悉BASH shell，那么本书中所出现的脚本将能够在你的系统上运行，因为BASH shell向后兼容Bourne shell。如果你使用Korn shell，那么你会发现，这两种shell 的语法非常接近。

如果你阅读一些安装程序，你会发现它们中的95%是用Bourne shell写的。这是因为写这些脚本的人知道用Bourne shell写的脚本能够在任何UNIX和LINUX系统上运行。

Shell的可移植性

编写shell脚本的时候，你希望你写出的脚本能够在任何系统上运行，这就叫做可移植性。

脚本的可移植性主要涉及两个方面:

- 所使用的脚本语法。
- shell命令。

如果打算使用Bourne shell编写脚本，那么第一个问题实际上就已经基本解决了：Bourne shell不存在不可移植性问题。

一个shell脚本的20%(或者很有可能更多)的部分使用到诸如cp、mv、mkdir等shell命令，这就是问题的所在。不同的系统供应商提供了不同的命令选项；如果你使用UNIX，BSD和System V版本，这个问题就更突出了。本书中所使用的脚本和命令例子具有通用性，基本上只用到了BSD UNIX和System V UNIX均有的命令及选项。凡是本书中所涉及到的在不同的系统上有微小的差别的命令，本书都指出了该命令的其他形式，不过，这种情况很少出现。

本书的章节安排

本书的章节安排兼顾学习工具和参考书的双重目的，因此在阅读本书时，不必按照章节顺序进行。如果你希望掌握基于cgi的网页设计的话，可以任意浏览其中的某些章节，哪怕是从最后一章看起。

本书分为五个部分，每一部分中的各个章节分别涉及shell和shell编程的各个不同方面。

书中含有大量有用的脚本例子。

第一部分　shell

本书的第一部分讨论了如何按照某种文件名模式来列出文件，如何快速地切换目录。当你创建一个文件时，你希望它具有正确的权限，存在于正确的目录中，这一部分就涉及了这样的内容。第一部分还涉及到诸如设置umask和改变文件的用户组所有权等文件操作。

为了便于搜索你所创建的文件，有专门的一章讨论find命令（参阅第2章）。

有时候出于某种考虑，你希望脚本在夜间运行：你可以在第3章“后台执行命令”中找到处理这一问题办法。

从文件或终端读入命令并显示输出是任何shell都应当具有的最基本功能。第一部分涉及了这一内容。有时你可能希望只有在前一个命令成功完成的情况下才运行某个命令—没问题，这一部分中也包含这样的内容。

第二部分　文本过滤

本书的第二部分涉及了所有的主要文字过滤工具。你的shell脚本在抽取信息、执行过程中以及在输出文本的时候都有可能涉及文字过滤，这一部分涵盖了所有需要掌握的相关内容。

在这里介绍了awk（第9章），它本身就是一种语言；grep（第8章），一种文字搜索工具；以及sed（第10章），一种易于使用的行编辑器。此外，还介绍了对文件和记录的排序、合并以及混排。还有一章用来讨论tr，一种字符转换工具（参阅第12章）。

第三部分　登录环境

第三部分讨论了登录环境以及如何定制登录环境。读了这一部分，你就能够知道在登录时哪些文件被执行；会理解所有的局部和全局环境变量；会对引用不再感到困惑，于是就能

够充分发挥环境变量的威力。

第四部分　基础shell编程

本书的第四部分全部都是关于shell脚本的。基本内容包括：如何使一个文件可执行，如何在shell中运行它；不同的控制结构和如何测试各种条件并依此结果采取不同的操作。此外，还对函数进行了讲解。函数是一段可重用的代码，你将看到如何使用函数，以及如何在不同的shell脚本之间共享这些函数。

如何向脚本传递参数是一个非常重要的问题。我们给出了三种向脚本传递参数的方法。

到现在为止我们已经可以编制实用的shell脚本，如果能够了解色彩和特殊控制字符的用法，岂不更好——猜猜看都有哪些特殊字符，我们将为你介绍这些内容。任何一本讲shell编程的书都会讲到文件的修改，在这部分中有专门的一章讨论这个话题。在这部分的最后，介绍了前面未涉及的一些内嵌shell命令。

第五部分　高级shell编程技巧

有没有更深入的shell编程技巧？当然有。本书的第五部分就是这样的内容。在这里，我们更深入地学习shell编程，探讨了信号和陷阱，这样，在某人试图杀死你的脚本进程时，这些进程就可以有所动作。我们还花费了一些笔墨来探讨<<的应用。你可能会问，<<是什么？可以使用它作为shell脚本的一种输入手段，从另外一种角度看，它可以作为一种从外部控制脚本运行的方法。

为什么有些程序能够在系统启动时自动被运行？你也许对此感到迷惑。这并不神秘，在这里我们讲述了运行级别的概念，教你如何构造运行级别脚本(rc.script)。按照书中的例子，你就能够让自己编写的shell脚本在系统启动时运行。

在这一部分还有专门的一章给出了几个短小精悍的shell脚本的例子，其中包含一个无需修改/etc/passwd文件就能够限制用户对系统访问的例子。如果希望了解如何把网页链接在一起，如何向另一个网页发送信息，如何实时地更新网页，你将在这里看到使用Bourne shell而不是Perl语言编制的cgi脚本。

本书还带有一个附录，包含了若干常用的shell命令。

阅读本书的前提

我们这里假定阅读本书的人知道如何登录进一个shell，如何切换目录，如何使用文字编辑器。

如果希望验证第五部分所讲述的cgi脚本，应当在系统上安装web服务器，并且该服务器允许运行cgi脚本(尽管这不是阅读这部分内容的必要条件)。

本书中的约定

全书中有如下约定：

<Ctrl-key>　代表同时按住Ctrl键和key所制定的键。例如<Ctrl-o>就是指同时按住Ctrl键和字母o。

在本书中，你将发现这样的文本框：

如果是LINUX系统，那么.....

这样的文本框用于指出当前所讲述的命令在BSD/LINUX和系统V之间的差别。

书中的脚本在LINUX(Redhat)和AIX系统上进行了测试，而且其中的部分脚本还在Data Generals上进行了测试。

希望你能够喜欢这本书，不仅把它当作是学习的工具，还把它看成一本参考工具书，并且能够从中获得乐趣，有所收益。

如果你有任何意见或看法，哪怕只是一个问候，请给我发电子邮件，地址是dtansley@my-Deja.com。

英文原书书号：ISBN 0-201-67472-6。

目　录

第二部分 文本过滤

第三部分 登录环境

第四部分 基础shell编程

第五部分 高级shell编程技巧

第一部分 shell

第1章 文件安全与权限

为了防止未授权用户访问你的文件，可以在文件和目录上设置权限位。还可以设定文件在创建时所具有的缺省权限：这些只是整个系统安全问题中的一小部分。在这里我们并不想对系统安全问题的方方面面进行全面的探讨，只是介绍一下有关文件和目录的安全问题。

本章包含以下内容：

- 文件和目录的权限。
- setuid。
- chown和chgrp。
- umask。
- 符号链接。

创建文件的用户和他(她)所属于的组拥有该文件。文件的属主可以设定谁具有读、写、执行该文件的权限。当然，根用户或系统管理员可以改变任何普通用户的设置。一个文件一经创建，就具有三种访问方式：

1) 读，可以显示该文件的内容。

2) 写，可以编辑或删除它。

3) 执行，如果该文件是一个shell脚本或程序。

按照所针对的用户，文件的权限可分为三类：

1) 文件属主，创建该文件的用户。

2) 同组用户，拥有该文件的用户组中的任何用户。

3) 其他用户，即不属于拥有该文件的用户组的某一用户。

1.1 文件

当你创建一个文件的时候，系统保存了有关该文件的全部信息，包括：

- 文件的位置。
- 文件类型。
- 文件长度。
- 哪位用户拥有该文件，哪些用户可以访问该文件。
- i节点。
- 文件的修改时间。
- 文件的权限位。

让我们使用ls -l命令，来看一个典型的文件：

```
$ ls -l
total 4232
-rwxr-xr-x   1 root   root      3756  Oct 14 04:44 dmesg
-r-xr-xr-x   1 root   root      12708 Oct  3 05:40 ps
-rwxr-xr-x   1 root   root      5388  Aug  5  1998 pwd
....
```

下面让我们来分析一下该命令所得结果的前面两行，看看都包含了哪些信息:

total 4232: 这一行告诉我们该目录中所有文件所占的空间。

-rwxr-xr-x: 这是该文件的权限位。如果除去最前面的横杠，这里一共是9个字符，他们分别对应9个权限位。通过这些权限位，可以设定用户对文件的访问权限。这9个字符可以分为三组:

rwx: 文件属主权限　　这是前面三位

r-x: 同组用户权限　　这是中间三位

r-x: 其他用户权限　　这是最后三位

后面我们还将对这些权限位作更详细的介绍。出现在r、w、x位置上的横杠表示相应的访问权限被禁止。

1 该文件硬链接的数目。

root 文件的属主。

root 文件的属主root所在的缺省组(也叫做root)。

3578 用字节来表示的文件长度，记住，不是K字节!

Oct 14 04:44 文件的更新时间。

dmesg 文件名。

1.2 文件类型

还记得前面一节所提到的文件权限位前面的那个字符吗？我们现在就解释一下这个横杠所代表的意思，文件类型有七种，它可以从ls -l命令所列出的结果的第一位看出，这七种类型是:

d 目录。

l 符号链接(指向另一个文件)。

s 套接字文件。

b 块设备文件。

c 字符设备文件。

p 命名管道文件。

- 普通文件，或者更准确地说，不属于以上几种类型的文件。

1.3 权限

让我们用touch命令创建一个文件:

```
$ touch myfile
```

现在对该目录使用ls -l命令:

```
$ ls -l
-rw-r--r--   1 dave     admin          0 Feb 19 22:05 myfile
```