

Linux版本。随着越来越多的软件公司开始销售自制版本的Linux，而且越来越多的硬件公司承诺在他们出售的计算机上预装Linux，自由软件与商业软件之间的界限变得愈发模糊了。

作为Linux故事的一个有趣的脚注，我们注意到在Linux变得越来越流行时，它从一个意想不到的源头（AT&T）获得了很大的推动。1992年，由于缺乏资金，Berkeley决定在推出BSD的最终版本4.4BSD后停止开发（4.4BSD后来成为FreeBSD的基础）。由于这个版本几乎不包含AT&T的代码，Berkeley决定将这个软件的开源许可证（不是GPL）发布，任何人可以对它做任何想做的事情，只要不对加州大学提出诉讼。AT&T负责UNIX的子公司做出了迅速的反应——正如你猜的那样——它提出了对加州大学的诉讼。同时，它也控告了BSDI，一家由BSD开发者创立、包装系统并出售服务的公司（正像Red Hat以及其他公司现在为Linux所做的那样）。由于4.4BSD中事实上不含有AT&T的代码，起诉是依据版权和商标侵犯，包括BSDI的1-800-ITS-UNIX那样的电话号码。虽然这次诉讼最终在庭外和解，它把FreeBSD隔离在市场之外，却给了Linux足够的时间发展壮大。如果这次诉讼没有发生，从1993年起两个免费、开源的UNIX系统之间就会进行激烈的竞争：由处于统治地位的、成熟稳定且自1977年起就在学界得到巨大支持的系统BSD应对富有活力的年轻挑战者、只有两年历史却在个人用户中支持率稳步增长的Linux。谁知道这场免费UNIXES的战争会变成何种局面？

## 10.2 Linux概述

为了那些对Linux不熟悉的用户的利益，在这一节我们将对Linux本身以及如何使用Linux进行简单的介绍。几乎本节介绍的所有内容同样适用于所有与UNIX相差不多的UNIX衍生系统。虽然Linux有多个图形界面，但在这里我们关注的是在X系统的shell窗口中工作的程序员眼中的Linux界面。在随后的几节中，我们将关注系统调用以及它们是如何在内核中工作的。

### 10.2.1 Linux的设计目标

一直以来，UNIX都被设计成一种能够同时处理多进程和多用户的交互式系统。它是由程序员设计的，也是给程序员使用的，而使用它的用户大多都比较有经验并且经常参与（通常较为复杂的）软件开发项目。在很多情况下，通常是大量的程序员通过积极的合作来开发一个单一的系统，因此UNIX有广泛的工具来支持在可控制的条件下的多人合作和信息共享。一组有经验的程序员共同开发一个复杂软件的模式显然和一个初学者独立地使用一个文档编辑器的个人计算机模式有显著区别，而这种区别在UNIX系统中自始至终都有所反映。Linux系统自然而然地继承了这些设计目标，尽管它的第一个版本是面向个人电脑的。

好的程序员追求什么样的系统？首先，大多数程序员喜欢让系统尽量简单，优雅，并且具有一致性。比如，从最底层的角度来讲，一个文件应该只是一个字节集合。为了实现顺序存取、随机存取、按键存取、远程存取等而设计不同类型的文件（像大型机一样）只会碍事。类似地，如果命令

```
ls A*
```

的意思是列举出所有以“A”打头的文件，那么命令

```
rm A*
```

的意思就应该是删除所有以“A”打头的文件而不是删除文件名是“A\*”的那个文件。这个特性有时被称为最小惊讶原理。

有经验的程序员通常还希望系统具有较强的功能性和灵活性。这意味着一个系统应该具有较小的一组基本元素，而这些元素可有多种多样的组合方式来满足各种应用需要。设计Linux的一个基本指导方针就是每个程序应该只做一件事并且把它做好。因此，编译器不会产生列表，因为有其他的程序可以更好地实现这个功能。

最后，大多数程序员非常反感没用的冗余。如果cp可以胜任，那么为什么还需要copy？为了从文件f中提取所有包含字符串“ard”的行，Linux程序员输入

```
grep ard f
```

另外一种方法是让程序员先选择grep程序（不带参数），然后让grep程序自己宣布说“你好，我是grep，我在文件中寻找模式。请输入你要寻找的模式。”在输入一个模式之后，grep程序要求输入一个文件名。