

第10章 实例研究1: Linux

在前面的章节中，我们大体上学习了很多关于操作系统的原理、抽象、算法和技术。现在分析一些具体的操作系统，看一看这些原理在现实世界中是怎样应用的。我们将从Linux开始，它是UNIX的一个很流行的衍生版本，可以运行在各类计算机上。它不仅是高端工作站和服务器的主流操作系统之一，还在移动电话到超级计算机的一系列系统中得到应用。Linux系统也体现了很多重要的操作系统设计原理。

我们将从Linux的历史以及UNIX与Linux的演化开始讨论，然后给出Linux的概述，从而使读者对它的使用有一些概念。这个概述对那些只熟悉Windows系统的读者尤为有用，因为Windows系统实际上对使用者隐藏了几乎所有的系统细节。虽然图形界面可以使初学者很容易上手，但它提供了很少的灵活性而且不能使用户洞察到系统是如何工作的。

接下来是本章的核心内容，我们将分析Linux的进程与内存管理、I/O、文件系统以及安全机制。对于每个主题，我们将先讨论基本概念，然后是系统调用，最后讨论实现机制。

我们首先应该解决的问题是：为什么要用Linux作为例子？的确，Linux是UNIX的一个衍生版本，但UNIX自身有很多版本，还有很多其他的衍生版本，包括AIX、FreeBSD、HP-UX、SCO UNIX、System V Solaris等。幸运的是，所有这些系统的基本原理与系统调用大体上是相同的（在设计上）。此外，它们的总体实现策略、算法与数据结构也很相似，不过也有一些不同之处。为了使我们的例子更具体，最好选定一个系统然后从始至终地对它进行讨论。因为大多数读者相对于其他系统而言更容易接触到Linux，故我们选中Linux作为例子。况且除了实现相关的内容，本章的大部分内容对所有UNIX系统都是适用的。有很多书籍介绍怎样使用UNIX，但也有一些介绍其高级特性以及系统内核（Bovet和Cesati，2005；Maxwell，2001；McKusick和Neville-Neil，2004；Pate，2003；Stevens和Rago，2008；Vahalia，2007）

10.1 UNIX与Linux的历史

UNIX与Linux有一段漫长而又有趣的历史，因此我们将从这里开始我们的学习。UNIX开始只是一个年轻的研究人员（Ken Thompson）的业余项目，后来发展成价值数十亿美元的产业，涉及大学、跨国公司、政府与国际标准化组织。在接下来的内容里我们将展开这段历史。

10.1.1 UNICS

回到20世纪40~50年代，当时使用计算机的标准方式是签约租用一个小时的机时，然后在这个小时内独占整台机器。至少从这个角度，所有的计算机都是个人计算机。当然，这些机器体积庞大，在任何时候只有一个人（程序员）能使用它们。当批处理系统在20世纪60年代兴起时，程序员把任务记录在打孔卡片上并提交到机房。当机房积累了足够的任务后，将由操作员在一次批处理中处理。这样，往往在提交任务一个甚至几个小时后才能得到结果。在这种情况下，调试成为一个费时的过程，因为一个错位的逗号都会导致程序员浪费数小时。

为了摆脱这种公认的令人失望且没有效率的设计安排，Dartmouth学院与M.I.T发明了分时系统。Dartmouth系统只能运行BASIC，并且经历了短暂的商业成功后就消失了。M.I.T的系统CTSS用途广泛，在科学界取得了巨大的成功。不久之后，来自Bell实验室与通用电器（随后成为计算机的销售者）的研究者与M.I.T合作开始设计第二代系统MULTICS（MULTiplexed Information and Computing Service，多路复用信息与计算服务），我们在第一章讨论过它。

虽然Bell实验室是MULTICS项目的创始方之一，但是它后来撤出了这个项目，仅留下一位研究人员Ken Thompson寻找一些有意思的东西继续研究。他最终决定在一台废弃的PDP-7小型机上自己写一个精简版的MULTICS（当时使用汇编语言）。尽管PDP-7体积很小，但是Thompson的系统实际上可以工作并且能够支持他的开发成果。随后，Bell实验室的另一位研究者Brian Kernighan有点开玩笑地把它叫做