

会展示应用程序员如何通过 Linux 系统调用来使用多个进程。学完本章之后，读者就能够编写带作业控制的 Linux shell 了。同时，这里也会向读者初步展示程序的并发执行会引起不确定的行为。

- 第 9 章：虚拟内存。我们讲述虚拟内存系统是希望读者对它是如何工作的以及它的特性有所了解。我们想让读者了解为什么不同的并发进程各自都有一个完全相同的地址范围，能共享某些页，而又独占另外一些页。我们还讲了一些管理和操纵虚拟内存的问题。特别地，我们讨论了存储分配操作，就像标准库的 `malloc` 和 `free` 操作。阐述这些内容是出于下面几个目的。它加强了这样一个概念，那就是虚拟内存空间只是一个字节数组，程序可以把它划分成不同的存储单元。它可以帮助读者理解当程序包含存储泄漏和非法指针引用等内存引用错误时的后果。最后，许多应用程序员编写自己的优化了的存储分配操作来满足应用程序的需要和特性。这一章比其他任何一章都更能展现将计算机系统硬件和软件结合起来阐述的优点。而传统的计算机体系结构和操作系统书籍都只讲述虚拟内存的某一方面。
- 第 10 章：系统级 I/O。我们讲述 Unix I/O 的基本概念，例如文件和描述符。我们描述如何共享文件，I/O 重定向是如何工作的，还有如何访问文件的元数据。我们还开发了一个健壮的带缓冲区的 I/O 包，可以正确处理一种称为 short counts 的奇特行为，也就是库函数只读取一部分的输入数据。我们阐述 C 的标准 I/O 库，以及它与 Linux I/O 的关系，重点谈到标准 I/O 的局限性，这些局限性使之不适合网络编程。总的来说，本章的主题是后面两章——网络和并发编程的基础。
- 第 11 章：网络编程。对编程而言，网络是非常有趣的 I/O 设备，它将许多我们前面文中学习的概念(比如进程、信号、字节顺序、内存映射和动态内存分配)联系在一起。网络程序还为下一章的主题——并发，提供了一个很令人信服的上下文。本章只是网络编程的一个很小的部分，使读者能够编写一个简单的 Web 服务器。我们还讲述位于所有网络程序底层的客户端-服务器模型。我们展现了一个程序员对 Internet 的观点，并且教读者如何用套接字接口来编写 Internet 客户端和服务端。最后，我们介绍超文本传输协议(HTTP)，并开发了一个简单的迭代式 Web 服务器。
- 第 12 章：并发编程。这一章以 Internet 服务器设计为例介绍了并发编程。我们比较对照了三种编写并发程序的基本机制(进程、I/O 多路复用和线程)，并且展示如何用它们来建造并发 Internet 服务器。我们探讨了用 P、V 信号量操作来实现同步、线程安全和可重入、竞争条件以及死锁等的基本原则。对大多数服务器应用来说，写并发代码都是很关键的。我们还讲述了线程级编程的使用方法，用这种方法来表达应用程序中的并行性，使得程序在多核处理器上能执行得更快。使用所有的核解决同一个计算问题需要很小心谨慎地协调并发线程，既要保证正确性，又要争取获得高性能。