

Application Program Interface (API) [C Language]

这是一个标准。一些部分确实值得一读，特别是附录B，清晰阐述了为什么要这样做。参考标准的一个好处在于通过定义不会出现错误。例如，如果一个宏的名字中的排字错误贯穿了整个编辑过程，那么它将不再是一个错误，而成为一种正式标准。

Fusco, *The Linux Programmers' Toolbox*

这本书是为那些知道一些基本Linux知识，并且希望能够进一步了解Linux程序如何工作的中级读者们写的。该书假定读者是一个C程序员。

Maxwell, *Linux Core Kernel Commentary*

该书的前400页给出了Linux的内核源代码的一个子集。后面的150页则是对这些代码的评述。与John Lions的经典书籍(1996)风格很相似。如果你想了解Linux内核的很多细节，那么这是一个不错的起点，但是读40 000行C语言代码不是每个人都必需的。

14.1.11 Windows Vista

Cusumano and Selby, "How Microsoft Builds Software"

你是否曾经好奇过一个人如何能够写出29 000 000行代码(就像Windows 2000一样)，并且让它作为一个整体运转起来?希望探究微软是如何采用建造和测试循环来管理大型软件项目的读者，可以参看这篇论文。其过程相当有启发性。

Rector and Newcomer, *Win32 Programming*

如果想找一本1500页的书，告诉你如何编写Windows程序，那么读这本书是一个不错的开始。它涵盖了窗口、设备、图形输出、键盘和鼠标输入、打印、存储管理、库和同步等许多主题。阅读这本书要求读者具有C或者C++语言的知识。

Russinovich and Solomon, *Microsoft Windows Internals, 4th ed.*

如果想学习如何使用Windows，可能会有几百种相关的书。如果想知道Windows内部如何工作的，本书是读者最好的选择。它给出了很多内部算法和数据结构以及可观的技术细节。没有任何一本书可以替代。

14.1.12 Symbian操作系统

Cinque et al., "How do Mobile Phone Fail? A Failure Data Analysis of Symbian OS Smart Phones"

以前不论怎样，当计算机崩溃时，至少电话总是可以用的。而现在电话其实就是一个小屏幕的计算机，它们也会因为糟糕的软件而崩溃。本文讨论了可以导致Symbian手机或者终端崩溃的软件错误。

Morris, *The Symbian OS Architecture Sourcebook*

如果你一直在寻找关于Symbian操作系统的进一步细节，那么本书是一个很好的开始。它涉及了Symbian的体系结构和相当数量的各层细节，而且还给出了一些实例分析。

Stichbury and Jacobs, *The Accredited Symbian Developer Primer*

如果你对为Symbian手机或者掌上电脑如何开发应用软件感兴趣的话，那么本书是一个不错的选择。它从所需要的C++语言讲起，逐步深入到系统结构、文件系统、网络管理、工具链和兼容性。

14.1.13 设计原则

Brooks, *The Mythical Man Month: Essays On Software Engineering*

Fred Brooks是IBM的OS/360的主要设计者之一。以其丰富的经验，他知道在计算机的设计中什么是可以运行的和什么是不能运行的。他在25年前写下这本诙谐且内涵丰富的书中给出的建议现在一样是可行的。

Cooke et al., "UNIX and Beyond: An Interview with Ken Thompson"

设计一个操作系统与其说是一门科学，不如说是一门艺术。因此，倾听该领域专家的谈话是一个学习这方面知识的有效途径。在操作系统领域中，没有谁比Ken Thompson更有发言权的了。在对这位UNIX、Inferno、Plan9操作系统的合作设计者的访问过程中，Ken Thompson阐明了在这个领域中我们从哪里开始和即将走向哪里等问题。

Corbató, "On Building Systems That Will Fail"

在获得图灵奖的演讲大会上，这位分时系统之父阐述了许多Brooks在《人月神话》中同样关注的问