

名为POSIX。前三个字母代表可移植操作系统 (Portable Operating System)，后缀IX用来使这个名字与UNIX的构词相似。

经过一次又一次的争论与辩驳之后，POSIX委员会制定了一个称为1003.1的标准。它规定了每一个符合标准的UNIX系统必须提供的库函数。大多数库函数会引发系统调用，但也有一些可以在系统内核之外实现。典型的库函数包括open，read与fork。POSIX的思想是这样的，一个软件销售商写了一个只调用了符合1003.1标准函数的程序，那么他就可以确信这个程序可以在任何符合标准的UNIX系统上运行。

的确大多数标准制定机构都会做出令人厌恶的妥协，在标准中包含一些制定这个标准的机构偏好的一些特性。在这点上，考虑到制定时牵涉到的大量相关者与他们各自既定的喜好，1003.1做得非常好。IEEE委员会并没有采用System V与BSD特性的并集作为标准的起始点（大部分的标准组织常这样做），而是采用了两者的交集。非常粗略地说，如果一个特性在System V与BSD中都出现了，它就被包含在标准中，否则就被排除出去。由于这种做法，1003.1与System V和BSD两者的共同祖先UNIX第7版有着很强的相似性。1003.1文档的编写方式使得操作系统的开发者与软件的开发者都能够理解，这是它在标准界中的另一个创新之处，即使这方面的改进工作已经在进行之中。

虽然1003.1标准只解决了系统调用的问题，但是一些相关文档对线程、应用程序、网络及UNIX的其他特性进行了标准化。另外，ANSI与ISO组织也对C语言进行了标准化。

10.1.6 MINIX

所有现代的UNIX系统共有的一个特点是它们又大又复杂。在这点上，与UNIX的初衷背道而驰。即使源代码可以免费得到（在大多数情况下并不是这样），单纯一个人不再能够理解整个系统。这种情况导致本书的作者编写了一个新的类UNIX系统，它足够小，因而比较容易理解。它的所有源代码公开，可以用作教学目的。这个系统由11 800行C代码以及800行汇编代码构成。它于1987年发布，在功能上与UNIX第7版几乎相同，后者是PDP-11时代大多数计算机科学系的中流砥柱。

MINIX属于最早的一批基于微内核设计的类UNIX系统。微内核背后的思想是在内核中只提供最少的功能，从而使其可靠和高效。因此，内存管理和文件系统被作为用户进程实现。内核只负责进程间的信息传递。内核包含1600行C代码以及800行汇编代码。由于与8088体系结构相关的技术原因，I/O设备驱动（增加2900行C代码）也在内核中。文件系统（5100行C代码）与内存管理（2200行C代码）作为两个独立的用户进程运行。

由于高度模块化的结构，微内核相对于单核系统有着易于理解和维护的优点。同时，由于一个用户态进程崩溃后造成的损害要远小于一个内核组件崩溃后造成的损害，因此将功能代码从内核移到用户态后，系统会更加可靠。微内核的主要缺点是用户态与内核态的额外切换会带来较大的性能损失。然而，性能并不代表一切：所有现代的UNIX系统为获得更好的模块性在用户态运行X-windows，同时容忍其带来的性能损失（与此相反的是Windows，其中整个GUI运行在内核中）。在那个时代，其他的著名微内核设计包括Mach（Accetta等人，1986）和Chorus（Rozier等人，1988）。

在问世几个月之内，MINIX在自己的USENET（现在的Google）新闻组comp.os.minix以及超过40 000名使用者中风靡一时。很多使用者提供了命令和其他用户程序，MINIX从而变成了一个由互联网上的众多使用者完成的集体项目。它是之后出现的其他集体项目的一个原型。1997年，MINIX第2版发布，其基本系统包含了网络，并且代码量增长到了62 200行。

2004年左右，MINIX发展方向发生了巨大的变化，它聚焦到发展一个极其可靠、可依赖的系统，能够自动修复自身错误并且自恢复，即使在可重复软件缺陷被触发的情况下也能够继续正常工作。因此，第1版中的模块化思想在MINIX 3.0中得到极大扩展，几乎所有的设备驱动被移到了用户空间，每一个驱动作为独立的进程运行。整个核心的大小突然降到不到4000行代码，因此一个单独的程序员可以轻易地理解。为了增强容错能力，系统的内部机制在很多地方发生了改变。

另外，超过500种流行的UNIX程序被移植到MINIX 3.0，包括X Window系统（有时候只用X代表）、各种各样的编译器（包括gcc）、文本处理软件、网络软件、浏览器以及其他很多程序。与以前的版本在本质上主要是教学用途不同，从MINIX 3.0开始拥有高可用性，并聚焦在高可靠性上。MINIX的最终目标是：取消复位键。