

Linux与Unix二大操作系统编程的安全 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_Linux_E4_B8_8EUn_c103_145161.htm 在考虑操作系统的继承安全性的时候，有一个常常被忽略了的问题（这到符合微软的口味）：如果你为Windows平台编程，那么你的软件就只能够在有限的操作系统、版本和补丁层上运行。而UNIX和Linux则相反，它们包括很多开放源代码、商业，以及政府开发的版本，而且应用程序常常跨很多不同平台自由发布，而这些平台都有相当不同的安全特性。由于有些读者只在Windows上工作，所以这里要介绍一下*nix。UNIX和Linux的划分在于其内核以及用户空间上。大多数程序都可以在用户空间里的内核之上运行。刚刚接触Linux的程序员，尤其是有经验的UNIX开发人员，有时候无法意识Linux并不只是是免费版本的UNIX；事实上，两者的确存在一些很大的不同之处。从安全的角度讲，大多数版本Linux都非常相似，因为它们都基于相同的内核和C库。一个显著的不同之处是，Linux对按需动态加载内核模块提供了专门的支持。除了UNIX和Linux以上的不同之处，不同版本的UNIX之间也存在很大的安全差别。对于转到UNIX的Linux开发人员来说，这些差别不太明显，但是很重要。例如，有些和UNIX类似的操作系统会进行强制的访问控制，而有的则不会。还有一个紧密相关的平台：Free Software HURD宏内核的*nix，它有自己的安全特性。有些UNIX平台，其中最显著的是OpenBSD，在进行缺省安装的时候本身就要比其它的操作系统更加安全；如果为它们所写的程序还会运行在其他一些版本的UNIX系统上，那么这

些应用程序仍然需要实现它们自己的安全特性，并遵照一些好的编程惯例。对于开发人员而言，理解线程和进程是尤其重要的。在UNIX的变体里，线程常常会被单独处理，而执行则由系统来计划安排。而在Linux里，进程和线程事实上没有区别。在UNIX里，同一个进程里的线程会共享内存，但是Linux的进程会创建另外的进程，并能够定义哪些资源可以被共享，而哪些不能。Linux的开发人员常常会用到“任务（task）”这个词，而他们事实上是在谈论进程。为了以最安全的方式让应用程序能够被移植，开发人员应该坚持使用标准的线程库。特别的，如果你希望应用程序能够被移植，那么就要使用“fork”调用来创建新的进程。要注意，如果可能的话要坚持使用fork，这一点很重要。不要使用“vfork”，它是由BSD开发的一个优化版本的fork（因此被广泛使用），而且出于兼容性的考虑，也被引入了Linux。其原因很简单：vfork会让子进程共享（或者更准确地说，接管）父进程的内存和控件，这会导致子进程和父进程之间的相互冲突。在有些版本的Linux里，vfork在特定情况下会被黑客攻破。尽管Windows有这样或者那样明显的和著名的问题，从某种角度上讲，微软的开发人员实现编程安全要相对容易一些，因为他们所工作平台的多样性受到了限制。UNIX和Linux的开发人员，尤其是跨平台的开发人员，可能对安全具有特别的敏感性，因为它们的操作系统可能在缺省的条件下就更加安全；但是，不同版本的UNIX之间在安全上有很多的不同，其差别甚至要比UNIX和Linux之间的还要大。它们之间众多的相同特性常常掩盖掉了这些差别，而可移植性的因素则加剧了这些差别。没有哪个Windows下的C或者VBASIC程序员，

会认为他的或者她的应用程序会一成不变的只运行在UNIX平台上，或者相反；但是在UNIX / Linux社区内部，要记住，由于应用程序可能会运行在不同的平台上，所以它可能无法运行在相同的安全级别上，这一点很重要。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com