

Linux 与 Windows 在桌面领域的互操作性^①

王亚军

(中国人民武装警察部队学院, 廊坊 065000)

摘要: Linux 和 Windows 都是计算机操作系统领域中的技术热点, 两者实现互操作有利于在竞争中共同推动操作系统技术向纵深发展。针对 Linux 和 Windows 在桌面领域中的互操作问题, 文中作了系统的阐述。在桌面领域, Linux 和 Windows 可以采用 API 仿真等虚拟化技术或修改操作系统源代码的方法来互运行对方程序, 可以采用开放、通用与统一的数据文件格式标准来互处理数据文件, 可以采用内核空间文件系统机制或用户空间文件系统机制来互访问文件系统等。

关键词: 操作系统; 互操作性; 兼容内核; 虚拟化; XML; 文件系统

Interoperability of Linux and Windows in Desktop Domain

WANG Ya-Jun

(Chinese People's Armed Police Forces Academy, Langfang 065000, China)

Abstract: Linux and Windows are both hot points in the academic field of computer technology. The interoperability of them two further expands their respective development area. On the other hand, some new technology problems are also produced. Focusing on the interoperability of Linux and Windows in desktop domain, this paper goes into a little more detail about these problems and their solutions. In desktop domain, the two operating systems can mutually run programs by adopting virtualization technologies such as API emulation or by altering source codes of operating systems, can mutually deal with data files by adopting open, common and unified data file format standards, and can mutually access file systems by adopting the mechanism of file system in kernel space or in user space.

Keywords: operating system; interoperability; unified kernel; virtualization; XML; file system

而今, 作为最成功的开源软件和最成功的商业软件, Linux 和 Windows 在操作系统应用领域中的二元主流格局基本形成^[1]。为了在两者之间架起桥梁, 两者之间的互操作问题逐渐成为人们关注的热点。桌面领域作为现代操作系统的主流应用领域, 虽然仍是 Windows 居于霸主地位, 但各种桌面 Linux 版本也相继推出, 在很大程度上改善了用户体验, 丰富了应用软件, 扩大了驱动支持, 与 Windows 的差距正在逐步缩小。研究这两种操作系统在桌面领域的互操作问题, 一方面有利于扩展 Linux 的桌面应用空间, 另一方面也为两者在网络领域、嵌入式领域的互操作技术打下坚实的基础, 从而推动操作系统互操作技术的全面发展^[2,3]。

从本质上说, Linux 和 Windows 是异构的操作系统。Linux 采用了高效而又模块化的宏内核结构, 而 Windows 则采用了高度模块化的扩大的微内核结构。Linux 内核由进程管理、存储管理、文件管理、网络管理和进程间通信等组件构成, 由一大段内核代码实现内核各个组件的功能, 而具体的文件系统或设备驱动等也能以可加载模块的方式动态地载入内核, 内核运行效率较高。Windows 核心态组件包括: 核心、执行体、设备驱动、硬件抽象层和图形引擎。与纯粹的微内核系统不同, Windows 将许多微内核外的系统函数置于核心态运行, 以减少系统开销。Linux 和 Windows 都实现了核心态与用户态两个特权级别, 操作系统中至关重要的代码运行在核心态, 而用户程序运行在用

① 收稿时间:2011-05-19;收到修改稿时间:2011-06-10

户态, 用户程序可以通过系统调用机制来使用特定的内核服务^[4,5]。由于 Linux 和 Windows 在体系结构上存在差异, 所以在 Linux 和 Windows 之间实现互操作面临许多技术难题。下面从 Linux 与 Windows 互运行对方程序、互处理数据文件、互访问文件系统三个主要方面阐述两个操作系统在桌面领域的互操作问题。

1 Linux与Windows互运行对方程序

这两个操作系统互运行程序主要包括互运行设备驱动程序和应用程序两个方面。由于设备驱动程序是在操作系统内核态运行的软件模块, 而异构操作系统所提供的设备驱动框架是不同的, 所以要想互运行设备驱动程序, 就需要对操作系统内核代码进行修改, 以提供兼容环境^[6,7]。只有微软公司才能修改 Windows 内核源代码来兼容 Linux 设备驱动程序, 但由于运行在 Windows 系统上的设备驱动程序十分丰富, 所以微软公司也并不热衷于修改 Windows 内核来兼容 Linux 设备驱动程序。

而 Linux 环境下的设备驱动程序相对不足, 修改 Linux 内核兼容 Windows 设备驱动程序对 Linux 的拓展应用领域意义重大。Windows 设备驱动框架的主体是 Windows 内核中的 I/O 子系统, 作为这个框架的扩充, Windows 对具有特殊性的网络设备驱动定义了一个子框架, 即网络驱动程序接口规格 NDIS(Network Driver Interface Specification)。NDIS 的功能是由 Windows 的一个内核动态安装模块 ndis.sys 实现的。符合 NDIS 接口标准的微端口驱动和中间层模块都可以在 ndis.sys 的支持下运行^[7,8]。开源软件 NDIS Wrapper 就是 ndis.sys 从 Windows 内核到 Linux 内核的移植, 实现了 Windows 网络接口设备驱动程序在 Linux 系统上的运行^[9]。如果要让 Linux 系统广泛支持各种 Windows 设备驱动程序, 就需要修改 Linux 内核, 在内核中实现 Windows 设备驱动的完整框架和 Windows 的设备驱动支撑界面, 目前这项工作还处于研究阶段^[10]。

与设备驱动程序不同, 应用程序是在操作系统用户空间运行的软件, 一般使用应用程序编程接口 API(Application Programming Interface)来完成其任务。Linux 与 Windows 是异构的操作系统, 两者所提供的 API 是不同的, 这就导致两者不能直接运行对方的应用程序。在不修改本地操作系统与异地应用程序源代码的前提下,

为了给跨操作系统运行的二进制可执行程序提供其正确运行所必需的 API 调用, 可以采用 API 仿真技术(即程序库级虚拟化技术)在本地操作系统内核与异地应用程序的进程之间(在本地操作系统的用户空间)创建一个 API 仿真服务进程, 该进程通过进程间通信机制负责将异地应用程序的进程所需的异地 API 调用转化为本地 API 调用, 其实质就是跨进程的系统调用^[10]。开源软件 Wine 是目前最成功的在 Linux 系统上运行 Windows 应用程序的 API 仿真软件, 已能仿真大部分 Windows API 和运行大部分 Windows 应用程序。Linux 程序员还可以借助 Wine 所提供的兼容 Windows API 的函数库 Winelib, 对一些 Windows 应用程序的源代码不作修改或作少量修改后, 在 Linux 系统中编译生成 ELF 格式的可执行文件或链接文件^[11]。开源软件 Cygwin 则运行于 Windows 操作系统上, 可以利用自身基于 Windows API 实现的包含 Linux API 的函数库 cygwin1.dll 对部分 Linux 软件进行编译使之能够在 Windows 系统上运行, 但不支持 Linux 二进制可执行程序在 Windows 系统上直接运行^[12]。

由于跨进程的系统调用至少增加两次进程调度与切换, 所以会带来一定的时间延迟和性能损失。而且许多系统调用是粒度很大的“宏”操作, 有时难以高效地实现某些微妙的机制, 甚至难以贴切地翻译另一组系统调用, 以至于某些应用程序在跨系统运行时会出现不好的效果。另外, 有些应用程序是与特定的设备驱动模块配套运行的, 在异地操作系统未能提供相应设备驱动的情况下, API 仿真服务进程无法支持该程序跨系统运行。这些局限性是 API 仿真技术无法克服的, 而只能通过修改操作系统的方法加以解决^[10]。开源软件 Longene 是一个由浙大网新资助的操作系统项目, 目的是把 Linux 内核扩充成一个既支持 Linux 应用程序和设备驱动程序也支持 Windows 应用程序和设备驱动程序的“兼容内核”^[10]。为此, 需要广泛借鉴 Wine^[11]、ReactOS^[13,14]、NDIS Wrapper^[9]及 Kernel-Win32^[10]等开源软件的研发成果, 在 Linux 内核中扩充“一个框架、两个界面”, 即 Windows 设备驱动框架、Windows 设备驱动支撑界面、Windows 系统调用界面。目前该软件的研发还处于起步阶段^[10]。免费软件 SFU(microsoft windows Services For Unix)是微软公司推出的旨在使得 Windows 系统能够兼容 Unix(及类 Unix)操作的工具集。SFU 利用位于 Windows 内核之上的与 Win32 子系统并列的 Interix 子系统对 Unix 及

Linux 等应用程序向 Windows 系统迁移、重新编译然后运行提供较高效较全面的支持。Interix 子系统向 Unix 及 Linux 等应用程序提供了丰富的 API 调用而不存在 API 仿真技术中的性能损失问题, 保证迁移后的程序能够高效地运行。SFU 工具集在 Windows 系统上营造了较完整的 Unix 环境, 使得伴随着应用程序的迁移而出现的支持应用程序运行的资源(如文件、打印、账户和数据库等)的迁移问题得到较好的解决^[5,15]。

此外, 虚拟化技术也可用于 Linux 和 Windows 之间互运行应用程序。通过在宿主操作系统上运行虚拟机监视器 VMM(Virtual Machine Monitor)软件(也叫虚拟机管理程序 Hypervisor)营造出一个虚拟机环境, 并在该虚拟机上运行客户操作系统, 这样就可以实现 Linux 和 Windows 之间互运行应用程序^[16]。商业软件 VMWare^[17]和开源软件 VirtualBox^[18]等就是基于这一方案实现的。这种方案不需要对操作系统源代码做任何修改, 但是 VMM 在截获并处理客户操作系统的指令时会造成性能上的损失。如果将这些影响性能的代码集成到客户操作系统中, 即少量修改客户操作系统源代码使其主动与 VMM 协同工作, 就可以大幅提高虚拟化的性能。开源软件 Xen 就是基于这种原理来实现 Linux 和 Windows 之间互运行应用程序(在借助 Intel VT-x 或 AMD Pacifica 硬件辅助虚拟化技术的前提下, 可以不修改客户操作系统的源代码)^[19,20]。如果根据客户操作系统的具体情况进一步修改其源代码将整个 VMM 融入客户操作系统并进行具体优化, 将得到一个无需显式借助某种 VMM 软件即可直接在宿主操作系统上运行的客户操作系统并进一步提高虚拟化的性能。开源软件 Colinux^[21]、Andlinux^[22]和 Lina^[23]等就是基于这种原理通过修改 Linux 内核实现的可以直接在 Windows 系统上运行的操作系统。相应地, 也可以通过修改宿主操作系统的方法来提供虚拟机功能以支持客户操作系统。Linux-2.6.20 内核正式引入了基于内核的虚拟机 KVM (Kernel-based Virtual Machine)技术, 该技术通过字符串设备/dev/kvm 管理被虚拟的硬件, 使用修改过的 Qemu 进程作为客户操作系统接口, 并为客户操作系统的运行新增了客户运行模式。利用 KVM 技术, Windows 操作系统可以无需修改地运行于 Linux 内核支持的虚拟机上, 从而获得更高的虚拟化性能。目前, Linux 内核中的 KVM 技术还在进一步完善之中^[24]。

2 Linux与Windows互处理数据文件

从技术上讲, Linux 与 Windows 之间互处理数据文件(如打开或编辑数据文件等)比互处理程序文件(如运行程序文件等)要容易一些。互处理数据文件的难点在于数据文件格式标准的开放、通用与统一, 而不在于数据文件格式转换器的研发^[25]。(类型不同的信息当然需要不同的应用软件采取不同的数据文件格式来表示和存储。同时, 由于用户需求的多样和系统环境的差异, 即使是类型相同的信息, 也常常通过不同的应用软件采取不同的数据文件格式来表示和存储^[26]。)

许多数据文件格式是开放的, 甚至是由相关国际组织制定或认可的。开发处理这些格式开放的数据文件的商业软件或开源软件, 在 Windows 和 Linux 平台上都是可行的。而有些数据文件格式却是封闭的, 特别是一些由大型软件公司制定并普及的格式(封闭格式有利于维护他们的商业利益, 如同封闭他们的软件源代码一样)。别人想要开发能够处理这些格式封闭的数据文件的商业软件或开源软件(包括转换器), 是很困难的, 往往达不到百分之百的兼容效果。随着 Linux 操作系统的迅速推广, 越来越多的软硬件供应商除了针对 Windows 平台之外还针对 Linux 平台提供应用软件和驱动软件。这就使得某些先前只能在 Windows 平台上处理的格式封闭的数据文件, 逐渐也能在 Linux 平台上得到处理^[27]。

在类型众多的应用软件和数据文件中, 办公软件(如文字处理软件等)及其处理的办公文档无疑是用户最多代表性最强的。除了微软公司垄断市场的基于 Windows 平台的 Microsoft Office 商业办公套件以外, 各种基于 Windows 或 Linux 平台的商业或开源的办公软件五花八门。为了让用户的办公文档独立于用户所使用的办公软件及系统平台从而使用户真正掌握文档信息的控制权, 一款由国际结构化信息标准促进组织 OASIS(Organization for the Advancement of Structured Information Standards)制定并维护的基于 XML 技术的开放文档格式 ODF (Open Document Format)标准于 2006 年被国际标准化组织 ISO (International Organization for Standardization)和国际电工委员会 IEC (International Electrotechnical Commission)接受为国际标准(ISO/IEC 26300-2006)并得到了广泛的支持^[28]。看到基于 XML 技术的开放文档规范势不可挡, 同时又不想失去自己在办公软件市场中的优势地位,

微软公司在部分开放了老版 Office 套件的二进制文档格式之外,又制定了新的基于 XML 技术的开放(但不通用)办公文件格式 OOXML(Office Open XML file formats) 标准,从 Office 2007 版开始推广应用,并于 2008 年被 ISO 和 IEC 接受为文档格式国际标准草案(ISO/IEC DIS 29500)^[29]。为了更好地满足中国用户的需求并摆脱文档技术标准受制于国外的被动局面,中国国家电子政务总体组所属的中文办公软件基础标准工作组组织国内诸多办公软件企业和科研院所制定了开放的基于 XML 技术的中文办公软件文档格式规范,称为统一办公文档格式 UOF(Unified/Uniform Office document Format),中文简称“标文通”(标准促进文档互通),并于 2007 年获批成为中国国家标准(GB/T20916-2007)^[30]。

这三大标准都采用了由万维网联盟 W3C(World Wide Web Consortium)制定的可扩展标记语言 XML(eXtensible Markup Language)技术,表明 XML 作为文档格式标准描述语言的国际地位已经确立。作为标准通用标记语言 SGML(Standard Generalized Markup Language)的一个优化子集,XML 继承了 SGML 的大部分优越性能并且大幅降低了 SGML 的复杂度,填补了超文本标记语言 HTML(HyperText Mark-up Language)的技术空白,不仅非常适合于日益复杂的万维网 WWW(World Wide Web)应用,而且被广泛应用于数据处理的诸多领域。相对于传统的二进制格式而言,采用纯文本格式的 XML 要消耗更多的数据存储空间,但实现了自描述性(自带 XML 文档类型定义)、可扩展性(作为元语言允许用户自定义各种标记和创建其它标记语言)、可校验性(支持应用软件进行结构校验)、互操作性(可描述和集成来自多种应用程序的数据)、跨平台性(支持不同计算机系统之间交换数据)、结构化(支持任意深层的嵌套结构)、国际化(支持世界上大多数文字)、数据内容与显示格式分离(同一数据可用不同的形式展现给不同的用户)、数据粒状更新(更新数据时不需要向服务器重传整个数据结构)、强大的超链接机制(允许利用文件结构定址到文件中的相对位置)、数据的本地计算和处理(不必由服务器处理)等优越性能。XML 语言本身与相关技术规范、高层应用协议等一起构成了 XML 标准体系,涵盖了 XML 文档模式技术(如 DTD 和 XML Schema 用来定义 XML 文档的逻辑结构)、XML 文档样式技术(如 CSS 和 XSL 用

来定义 XML 文档的显示样式)、XML 文档查询技术(如 XQuery 等 XML 文档查询语言)、XML 文档解析技术(如 DOM 和 SAX 用来解析 XML 文档的内容和结构)、XML 文档链接技术(如 XLink、XPointer 和 XPath 用来实现 XML 链接功能)等先进技术,从而成为制定办公文档格式标准的基础工具,有利于促进办公文档格式三大标准的互通与融合,在 Linux 与 Windows 互处理数据文件方面发挥重要作用^[31]。

3 Linux与Windows互访问文件系统

作为操作系统的重要组成部分,文件管理系统在 Linux 宏内核与 Windows 微内核中的实现机制存在较大差异。为了使 Linux 的用户程序能够通过一组统一的系统调用接口来操作各种不同的具体文件系统, Linux 在用户程序与各种具体文件系统之间提供了一个虚拟文件系统 VFS(Virtual File System, 或 Virtual Filesystem Switch)。各种具体文件系统以一组统一的内核数据结构集合作为与 VFS 的界面来封装各自的文件操作函数,这样工作在内核态的 VFS 就向用户程序隐藏了各种具体文件系统的实现细节。Linux 还通过文件系统安装与拆卸的方法,将格式化成各种具体文件系统并建立各自目录树的所有存储设备,统一为一个总的文件系统目录树结构,使得从整个文件系统唯一的总根目录出发根据给定的文件全名(文件所在路径全名+文件本身名)便可以找到文件系统中的任何一个文件^[4,32]。相比之下,Windows 文件系统目录树结构是以“卷”为基础的(一个磁盘可以分建多个卷,一个卷也可以跨越多个磁盘),每个卷都可以格式化成一种具体文件系统,都有自己的驱动器名和根目录,但在各卷之上没有一个总的根目录。Windows 并没有在各种具体文件系统之上建立一个虚拟文件系统,而是要求每种具体文件系统提供自己的文件系统驱动程序 FSD(File System Driver)。FSD 工作在内核模式,负责向 I/O 管理器注册自己,并接受 I/O 管理器的调动以完成具体文件系统的相关操作^[5,33]。总的来说, Linux 对文件系统的管理更加严谨,而 Windows 更关心实用。

由于 Linux 与 Windows 在文件管理系统上存在较大差异,两者互访问对方的默认具体文件系统面临一些技术问题。对这两个操作系统而言,在自己的内核代码中直接实现对方默认具体文件系统的各种操作,无疑是效率最高的。为了满足 Linux 文件

管理系统中 VFS 对具体文件系统的接口要求, Windows 的默认具体文件系统(FAT 系列和 NTFS 系列)必须经过封装,才能在 Linux 内核中得以具体实现。通常,这种封装需要在使用现场(on the fly)引入一些特殊处理(待封装的具体文件系统本身虽不包含索引节点却要根据 VFS 的要求在内存中装配索引节点结构体,本身将目录看作特殊对象却要根据 VFS 的要求将目录表示为文件形式等),使得 Windows 的默认具体文件系统能够融入 Linux 文件管理系统。完成与 VFS 的衔接之后,还要在 Linux 内核中编写代码实现 Windows 默认具体文件系统的实际操作。目前,在标准 Linux 内核代码中已经实现对 Windows 下 FAT 系列文件系统的可读写操作,而对 NTFS 系列文件系统只实现了可读操作和受限的写操作,主要是因为 NTFS 过于复杂,且微软公司并未公布 NTFS 驱动程序源代码和格式文档,开发者只能做逆向工作从分析磁盘二进制数据入手研究 NTFS 的技术细节,并据此进行相关开发工作^[34]。鉴于编写内核代码支持具体文件系统的开发难度较大, Linux 内核也引入了用户空间文件系统 FUSE(File system in USEr space)机制,支持在用户空间实现具体文件系统操作。通过 Linux 内核模块 FUSE,一个用户空间文件系统可以被挂载为内核的一个额外文件系统组件。这种机制扩展了 Linux 内核对新型具体文件系统的支持,但会带来额外的内核态与用户态切换开销^[35]。开源软件 Ntfsmount^[36]和 NTFS-3g^[37]就是利用 FUSE 机制实现 Linux 操作系统对 NTFS 文件系统的可读写操作。此外,开源软件 Captive NTFS 另辟蹊径实现 Linux 对 NTFS 文件系统的可读写访问。它主要是在 Linux 用户空间利用 ReactOS 为 Windows 的 NTFS 驱动程序源代码文件 Ntfs.sys 的执行提供虚拟环境。但由于 Ntfs.sys 是商业软件, Captive NTFS 的应用受到限制。而且 Captive NTFS 的应用需要在 Linux 内核空间与用户空间频繁切换,所以效率低下^[38]。除了各种开源软件,商业软件 Paragon NTFS for Linux 也能实现 Linux 对 NTFS 文件系统的可读写访问^[39]。

反过来,在 Windows 中访问 Linux 默认具体文件系统(EXT 系列)也存在内核空间与用户空间两种实现途径。不同于普通的设备驱动程序, FSD 在 Windows 内核中需要在可安装文件系统 IFS(Installable File System)的框架下

开发,涉及到与 I/O 管理器、虚拟内存管理器 and 高速缓存管理器等的紧密交互^[5,33]。由于 Windows 文件管理系统的源代码并不开放,微软公司提供的关于 FSD 开发的帮助资料又非常简略,加之在内核空间编程本身就非常复杂,所以在 Windows 内核空间开发 EXT 系列 FSD 的难度较大。开源软件 Ext2fsd^[40]和免费软件 Ext2ifs^[41]等是在 Windows 内核空间开发的 EXT 系列 FSD,能够实现 EXT2 和 EXT3 文件系统的可读写访问。借鉴 Linux 等操作系统所引入的 FUSE 机制,在 Windows 中也可以开发 EXT 系列的用户空间文件系统。为此,需要在 Windows 内核中开发一个内核模式的 FSD 以充当普通用户程序与用户空间文件系统之间的代理程序。用户空间文件系统通过动态链接库向内核模式的 FSD 注册回调函数,内核模式的 FSD 通过调用回调函数来使用用户空间文件系统服务并将处理结果返回给普通用户程序。开源软件 Dokan 支持在 Windows 用户空间开发文件系统,目前该软件还在不断完善之中,未来在 Windows 可读写访问复杂的 EXT4 文件系统方面可能发挥重要作用^[42]。

4 结语

综上所述, Linux 与 Windows 在桌面领域的互操作性虽然已经取得了很多成果,但仍在不断发展完善之中。在运行程序方面,由浙大网新资助的 Linux 兼容内核项目(开源软件 Longene)正在稳步地向前推进,未来有望并入标准 Linux 内核中,使得标准 Linux 内核能够兼容运行 Windows 程序^[10]。在数据文件方面,国际上出现了基于 XML 技术的 ODF、OOXML 和 UOF 三种开放的办公文档格式,三种格式未来有望在博弈中归于统一,形成新的统一的办公文档格式国际标准^[43]。在文件系统方面,用户空间文件系统机制已经取代内核空间文件系统机制,成为 Linux 与 Windows 互访问对方默认文件系统的发展道路,未来有望实现互访问的完全可读写目标^[35,42]。随着开源软件运动在全球的蓬勃发展,开源运动已经成为我国软件发展的重要机遇,我们可以引进开源社区先进的开发机制,实现合作创新与自主创新相结合,建立起从操作系统到办公软件乃至整个基础软件体系的一整套中国独立自主的软件产业链^[44]。

参考文献

- 1 陆首群.Linux / Windows 二元主流并进.信息系统工程, 2007,(11):81-82.

- 2 Interoperability.<http://en.wikipedia.org/wiki/Interoperability>, 2010.
- 3 陆首群. 探讨开源软件的互操作策略. 软件世界, 2007, (6): 57-59.
- 4 毛德操, 胡希明. Linux 内核源代码情景分析. 杭州: 浙江大学出版社, 2002.
- 5 陈向群, 向勇, 王雷等. Windows 操作系统原理(第 2 版). 北京: 机械工业出版社, 2005.
- 6 Corbet J, Rubini A, Kro A, Hartman G. LINUX Device Drivers. Sebastopol, California: OREILLY & Associates INC, 2005.
- 7 Orwick P. Developing Drivers with the Windows Driver Foundation. Redmond, Washington, USA: Microsoft Press, 2007.
- 8 Russinovich ME, Solomon DA, Chin AJ. Microsoft Windows Internals Redmond, Washington: Microsoft Press, 2005.
- 9 NDISwrapper.<http://wiki.debian.org/NdisWrapper>, 2010.
- 10 毛德操. 关于 Longene 开发: 漫谈兼容内核: 漫谈 Wine.<http://www.longene.org/whitebook.php>, 2010.
- 11 About Wine.<http://www.winehq.org/about/>, 2010.
- 12 Cygwin.<http://www.cygwin.com/>, 2010.
- 13 ReactOS Development Whitepaper.http://www.reactos.org/en/dev_whitepaper.html, 2010.
- 14 毛德操. Windows 内核情景分析. 采用开源代码 ReactOS 北京: 电子工业出版社, 2009.
- 15 Services for UNIX.<http://www.microsoft.com/china/windowsserver2003/sfu/default.msp>, 2010.
- 16 英特尔开源软件技术中心. 复旦大学并行处理研究所. 系统虚拟化. 原理与实现. 北京: 清华大学出版社, 2009.
- 17 Ward B. The Book of VMware. Sebastopol, California, USA: O'Reilly & Associates INC, 2005.
- 18 Virtualbox technical documentation.http://www.virtualbox.org/wiki/VirtualBox_architecture, 2010.
- 19 石磊, 邹德清, 金海. Xen 虚拟化技术. 武汉: 华中科技大学出版社, 2009.
- 20 Xen.<http://www.xen.org/>, 2010.
- 21 What is coLinux.<http://www.colinux.org/>, 2010.
- 22 Andlinux introduction.<http://www.andlinux.org/>, 2010.
- 23 Lina-Linux Network Administrator.<http://lina.sourceforge.net/>, 2010.
- 24 Kernel-based Virtual Machine. http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine, 2010.
- 25 万小利, 孙萍. Windows 常见文件格式详解. 北京: 北京理工大学出版社, 2001.
- 26 姜楠, 王健. 常用多媒体文件格式与压缩标准解析. 北京: 电子工业出版社, 2005.
- 27 倪光南. 推行开放标准, 促进自主创新. 中国标准化, 2007, (2): 7-8, 11.
- 28 ISO/IEC 26300:2006.http://www.iso.org/iso/catalogue_detail.htm?csnumber=43485, 2010.
- 29 Office Open XML.http://en.wikipedia.org/wiki/Office_Open_XML#ISO.2FIEC_29500:2008, 2010.
- 30 UOF 国家标准文档格式研制过程.<http://www.uofsd.com/index.php?mod=news&act=shownews&id=34>, 2010.
- 31 贾素玲, 王强, 许珂, 等. XML 核心技术. 北京: 高等教育出版社, 2005.
- 32 Love R. Linux Kernel Development. New York, New York, USA: Macmillan Computer Pub, 2005.
- 33 潘爱民. Windows 内核原理与实现. 北京: 电子工业出版社, 2010.
- 34 Linux NTFS file system support.<http://sourceforge.net/projects/linux-ntfs/>, 2010.
- 35 Filesystem in Userspace.<http://fuse.sourceforge.net/>, 2010.
- 36 ntfsprogs.http://en.wikipedia.org/wiki/Ntfsprogs#Included_programs, 2010.
- 37 NTFS-3G Manual.<http://www.tuxera.com/community/ntfs-3g-manual/>, 2010.
- 38 Captive: The first free NTFS read/write filesystem for GNU/Linux.<http://www.jankratochvil.net/project/captive/>, 2010.
- 39 NTFS&HFS for Linux 8.1 Express.<http://www.paragon-software.com/home/ntfs-linux-per/index.html>, 2010.
- 40 Ext2Fsd Project FAQ.http://www.ext2fsd.com/page_id=7, 2010.
- 41 EXT2 IFS for Windows. <http://uranus.chrysocome.net/linux/ext2ifs.htm>, 2010.
- 42 Dokan.<http://dokan-dev.net/en/>, 2010.
- 43 倪光南. 关于文档格式标准等新建议.<http://www.blogchina.com/20080226481208.html>, 2008.
- 44 倪光南. 开源: 中国软件的出路. 软件世界, 2006, (14): 20-21.