Linux 与 Windows 互操作综述①

王亚军

(中国人民武装警察部队学院，廊坊 065000)

摘 要：针对 Linux 与 Windows 在桌面领域、网络领域和嵌入式领域的互操作问题，做了综合阐述。在桌面领域,两者可以互运行对方程序、互处理数据文件、互访问文件系统；在网络领域，两者可以采用共同的网络协议来支持对方系统中的资源与服务在网络环境下的共享操作；在嵌入式领域，两者可以采用虚拟化和代码重构等技术来支持对方应用软件在本系统中的交叉开发和向本系统的移植等。

关键词：操作系统；互操作性；兼容内核；虚拟化；文件系统；网络协议；嵌入式系统

Overview of the Interoperability of Linux and Windows

WANG Ya-Jun

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

Abstract: Aiming at the problems of interoperability between Linux and Windows in desktop domain, network domain and embedded domain, solutions are systematically illustrated in this paper. In desktop domain, the two operating systems can mutually run programs, can mutually deal with data files, and can mutually access file systems. In network domain, the two systems can support the shared operations of resources and services between them under the network environment by adopting the same network protocols. In embedded domain, by adopting the technologies such as virtualization and code refactoring, the two systems can mutually support the cross development of application softwares in local system, mutually support the transplanting of application softwares to local system.

Key words: operating system; interoperability; unified kernel; virtualization; file system; network protocol; embedded system

众所周知，Windows 是迄今为止在商业上最成功的操作系统，而 Linux 则是目前成长最快的操作系统。在全球范围内，两者在桌面领域、网络领域和嵌入式领域展开了激烈的竞争。在桌面领域，各种新版本的 Linux 系统相继推出，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，与居于霸主地位的 Windows 桌面系统的差距正在逐步缩小。在网络领域，Linux 凭借其成熟稳定的技术性能与 Windows 相抗衡。在嵌入式领域，代码开放、免费授权的 Linux 已经领先 Windows。可见，Linux 和 Windows 在操作系统应用领域中的二元主流格局基本形成。为了对两者进行优势互补，在实际应用中需要在这两个操作环境之间架起桥梁，即实现两者的互操作。

1 操作系统互操作技术

操作系统互操作技术是通过约定的接口或协议实现两个异构操作系统之间互换数据与共享服务的技术。主要包括：在桌面领域，相互支持对方程序在本操作系统中的运行，相互支持对方数据文件在本操作系统中的处理，相互支持对方文件系统在本操作系统中的访问等；在网络领域，相互支持对方系统中的资源与服务在网络环境下的共享操作等；在嵌入式领域，相互支持对方应用软件在本系统中的交叉开发和向本系统的移植等[1]。

① 基金项目:安徽省教育厅自然科学基金(2005KJ004ZD)
收稿时间:2011-07-16;收到修改稿时间:2011-09-05

目前，Windows 和 Linux 之间的互操作已成为人们关注的热点。从本质上说，Windows 和 Linux 是异构的操作系统。Windows 采用了高度模块化的扩大的微内核结构[2]，而 Linux 采用了高效而又模块化的宏内核结构[3]。另外，Windows 源代码不是开放的，因而我们只能在完全开源的 Linux 内核以及操作系统内核以外的应用层上寻求两个异构系统的互操作途径，这是研究两者互操作的难点问题。

2 Linux与Windows在桌面领域的互操作

Linux 与 Windows 互运行程序包括互运行设备驱动程序和应用程序两个方面。开源软件 NDIS Wrapper 将 ndis.sys 从 Windows 内核移植到 Linux 内核，实现了 Windows 网络接口设备驱动程序在 Linux 系统上的运行[4]。开源软件 Longene 则为修改 Linux 内核兼容各种 Windows 设备驱动程序制定了切实可行的技术路线[5]。API 仿真技术是一种程序库级虚拟化技术，通过在本地操作系统的用户空间创建一个 API 仿真服务进程，将对方应用程序运行所需的异地 API 调用转化为本地 API 调用，来兼容运行对方应用程序[5]。利用传统的虚拟化技术，在宿主操作系统上运行虚拟机监视器软件来支持客户操作系统，也可以实现 Linux 和 Windows 之间互运行应用程序[6]。Linux-2.6.20 内核正式引入了基于内核的虚拟机，使得 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[7]。免费软件 SFU 是微软公司推出的使得 Windows 系统能够兼容 Unix(及 Linux)操作的工具集，对 Unix(及 Linux)应用程序向 Windows 系统迁移(重新编译然后运行)提供较全面的支持。SFU 还通过使用网络文件系统 NFS 和远程登录工具 Telnet 等来支持 Windows 计算机与 Unix(及 Linux)计算机在网络领域实现互操作[8]。

Linux 与 Windows 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。对于格式封闭的数据文件，别人想要开发处理这些文件的应用软件，往往达不到百分之百的兼容效果。而对于格式开放的数据文件，在 Windows 和 Linux 平台上都可以开发处理这些文件的应用软件[9]。

Linux 与 Windows 互访问文件系统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。标准 Linux 内核空间已经实现对 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[10]。开源软件 Ext2fsd 等和免费软件 Ext2ifs 等则是在 Windows 内核空间开发的 Linux 默认文件系统(EXT 系列)的驱动程序，能够实现对 EXT2 和 EXT3 文件系统的可读写访问[11,12]。另一方面，Linux 内核引入了用户空间文件系统 FUSE(File system in USEr space)机制，开源软件 Ntfsmount 和 NTFS-3g 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[13,15]。开源软件 Dokan 则支持在 Windows 用户空间开发文件系统，在 Windows 可读写访问 EXT 系列文件系统方面发挥重要作用[16]。此外，开源软件 Captive NTFS 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 提供虚拟执行环境，实现 Linux 对 NTFS 文件系统的可读写访问[17]。

3 Linux与Windows在网络领域的互操作

计算机在网络中正确通信的基础是网络协议。网络中采用异构操作系统 Linux 与 Windows 的计算机，只要支持共同的网络协议，就可以相互通信，从而共享资源和服务，实现互操作。作为当前应用最广泛的网络协议栈，TCP/IP 协议栈在 Linux 与 Windows 中都得到了实现，自然成为两个操作系统在网络领域实现互操作的基础，许多支持 Linux 与 Windows 互操作的网络协议是基于 TCP/IP 协议栈实现的。利用 TCP/IP 协议栈中的超文本传输协议 HTTP(HyperText Transfer Protocol)、文件传输协议 FTP(File Transfer Protocol)和网络电传 TELNET(TELetype over the NETwork)等应用层协议就可以在 Linux/Windows 系统中构建相应的服务器，向 Windows /Linux 系统提供基本的资源共享服务。

为了在不安全的网络环境中提供较安全的远程登录等网络服务，互联网工程任务组 IETF 制定了安全外壳 SSH(Secure SHell)协议栈。该协议栈是以 TCP/IP 协议栈为基础的位于传输层和应用层之间的安全协议栈，主要包括安全传输层协议、用户认证协议和连接协议，采用了多种加密方式和认证方式，加强 TCP/IP 协议栈的安全性，并能以压缩方式传输数据以加快传输速度。在 Linux 与 Windows 系统中都可以利用 SSH 协议栈构建服务器向对方提供较安全的远程登录等网络服务。不仅如此，利用 SSH 协议栈，在 Linux 系统

中基于 FUSE 构建的 SSHFS 文件系统客户端程序可以将 Windows 服务器上的共享目录挂载到本地目录树中直接访问，而在 Windows 系统中基于 Dokan 构建的 SSHFS 文件系统客户端程序可以将 Linux 服务器上的共享目录挂载到本地目录树中直接访问。在访问过程中，数据通过 SSH 协议栈加密传输，安全而高效[18-20]。

在 Linux 与 Windows 系统中还可以基于各种能够显示远程桌面的协议构建服务器向对方提供基于图形界面的更直观友好的远程管理服务。虚拟网络计算 VNC(Virtual Network Computing)协议采用基于 TCP/IP 协议栈的远程帧缓冲 RFB 协议来进行远程图形界面控制[21]。远程桌面协议 RDP (Remote Desktop Protocol)是微软公司在国际电信联盟制定的 T.120 协议族基础上扩展设计而成的基于 TCP/IP 协议栈的网络协议。由于该协议是技术细节封闭的商用协议，为了在 Linux 系统上运用该协议构建服务器向 Windows 客户端提供远程显示服务，就需要采用基于地址解析协议 ARP 欺骗的中间人攻击等手段来实现[22]。X 协议是 X 窗口系统中 X 客户端与 X 服务器之间的通信协议。X 客户端与 X 服务器可以运行在同一台机器上，也可以借助 TCP/IP 协议栈运行于不同的机器上，为实施远程管理奠定基础[23]。独立计算架构 ICA (Independent Computing Architecture)协议是 Citrix 公司制定的基于服务器计算的网络协议，具有良好的平台无关性、协议无关性支持[24]。由于各种显示远程桌面的协议一般未提供安全性支持，所以在实践中常常结合使用基于 TCP/IP 协议栈的安全套接层 SSL(Secure Sockets Layer)协议[25]或前述的 SSH 协议等以加强网络通信的安全性。除了传统的客户机/服务器(Client/Server，C/S)模式以外,基于 HTTP 协议的浏览器/服务器(Browser/Server，B/S)模式也可用于 Linux 与 Windows 之间在网络领域相互进行远程管理，以满足简便灵活的管理需求[26]。

目前在局域网上为集成 Linux 与 Windows 计算机以相互共享文件和打印机等资源而广泛应用的协议是由微软等公司联合制定的服务器信息块 SMB(Server Message Block) 协议，广泛应用于 Windows 系统中。该协议是基于 TCP/IP 协议栈的使用网络基本输入/输出系统 NetBios 的较安全的网络协议。通过 NetBios 的名字服务器等，在 Linux 系统中利用 SMB 协议构建的服务器可以在 Windows 系统中的网上邻居里看到，Windows 客户端不需要更改设置就能如同使用 Windows 服务器一样使用 Linux 计算机上的资源[27]。为了将该协议从局域网推广到因特网中去，让因特网中的主机能够通过该协议共享文件和打印机等资源，微软公司对该协议进行了扩展，推出了开放的通用因特网文件系统 CIFS (Common Internet File System)协议，并提交到 IETF 以使它能够成为因特网上的一个标准协议[28]。

利用网络协议将远地目录挂载到本地目录树中供本地程序直接访问，对于本地程序来说无疑是最透明的访问方式。网络文件系统 NFS(Network File System)协议就是在因特网上异构计算机系统之间构建分布式文件系统的一个事实上的标准协议。该协议是基于 TCP/IP 协议栈采用远地过程调用 RPC 协议实现的无状态的较安全的网络协议。标准 Linux 内核支持 NFS，包括客户端与服务器端，其中客户端被置于 VFS 的管理之下，而服务器端需要创建几个守护进程来提供服务。标准 Linux 内核源代码实现的 NFS 运行效率较高，但有一定的局限性，为此出现了一些在 Linux 用户空间实现的 NFS 应用软件。Windows 利用 SFU 工具集或第三方软件提供 NFS 客户端与服务器端工具。在 Linux 与 Windows 系统中都可以利用 NFS 协议构建服务器向对方提供 NFS 服务，将远地目录挂载到本地目录树中供本地程序直接访问[29-32]。

在网络存储方面，基于 TCP/IP 协议栈传输 SCSI 命令和数据的因特网小型计算机系统接口 iSCSI(Internet Small Computer System Interface)协议已经被 IETF 接受为标准协议。在 Linux 与 Windows 系统中都可以利用该协议构建服务器在数据块级向对方提供有别于 SMB 和 NFS 协议的性优价廉的海量数据网络存储服务[33]。

总之，基于 TCP/IP 协议栈的各种网络协议从不同方面满足 Linux 与 Windows 在网络领域的互操作需求。由于具体应用多种多样，用户往往还需要根据具体情况编写网络应用程序在 Linux 与 Windows 系统间进行通信。在各种网络应用程序编程接口中，基于 TCP/IP 协议栈的套接字(Socket)是因特网上最通用的网络编程接口，可以实现跨 Linux 与 Windows 平台编写网络应用程序进行通信[34]。对于复杂的具体应用，用户甚至需要在 Linux 与 Windows 内核中开发新的网络协议。Linux 网络子系统具有清晰的内部界面和良好的可扩展性，在 Linux 内核中添加网络协议涉及到从数据链路层到传输层的修改，内容包括协议注册、协议相关操作、协议头实现、数据包接收、用户调用接口等方面[35]。Windows

为网络子系统定义了网络驱动程序接口规格 NDIS，跨越了数据链路层、网络层和传输层，将网络驱动程序由下向上分为网卡驱动程序、中间驱动程序和协议驱动程序三层结构。利用微软公司提供的开发工具包，遵循 NDIS 规范，就可以专注于具体协议的实现，而把层次之间的衔接工作交给 NDIS 去协调[36,37]。有了共同的网络协议，Linux 与 Windows 就可以实现网络互操作来满足具体应用的需求。

4 Linux与Windows在嵌入式领域的互操作

Linux 与 Windows 在嵌入式领域都取得了巨大的成功，但各具特色。Linux 内核精简而高效，可修改性强，支持多种体系结构，具有非常好的网络性能，而且 O(1)进程调度算法和内核空间可抢占机制等的引入大大提高了内核的实时性。嵌入式 Windows 与桌面 Windows 联系紧密。Windows XP Embedded 是桌面 Windows XP 专业版的组件化版本，两者是二进制兼容的。Windows Embedded CE 虽是一个全新设计的微内核嵌入式操作系统，但其提供的应用程序编程接口是 Win32 API 的一个子集。微软公司向开发者提供 Windows Embedded CE 的源代码，以提高嵌入式 Windows 的市场竞争力。

由于嵌入式硬件系统资源有限，嵌入式环境下的软件开发往往要采用宿主机/目标板的模式。通常情况下，宿主机和目标板上所采用的桌面操作系统和嵌入式操作系统属于同一类型。但是借助于虚拟化工具，开发者也可以在自己熟悉的一类桌面操作系统(桌面 Linux 或桌面 Windows)中为另一类嵌入式操作系统(嵌入式 Windows 或嵌入式 Linux)构建交叉开发环境，或将两类操作系统整合为一个开发平台以使两类开发工具协同工作[38]。各种虚拟化软件(如 Wine、Cygwin、VMWare、Colinux 等)都可以用于这一目的。而采用异类操作系统的宿主机和目标板之间还可以采用双方都支持的 FTP、TELNET、SSH、SMB、NFS 等网络协议进行数据传输。

由于嵌入式硬件系统资源有限，在嵌入式 Linux 与嵌入式 Windows 操作系统之间进行动态链接库(或叫共享对象)和应用软件的移植往往只能采用代码重构的方法，对软件源代码进行二次设计使其符合目标系统的运行环境。动态链接库和应用软件都运行于操作系统的用户空间，应用软件一般通过动态链接库进行系统调用以取得内核服务。异构操作系统具有不同的系统调用和操作规则，在嵌入式 Linux 与嵌入式 Windows 之间移植动态链接库首先要解决相关系统调用和操作规则的转化问题。动态链接库在异构操作系统中往往采用不同版本的 C 语言编写，涉及不同的头文件，不同的关键字，不同的数据类型，不同的编译文件，不同的调试环境等，移植时要对相关内容进行修改[39]。而移植嵌入式应用软件在实践中常采用逆向逐步代码重构法或逆向总体代码重构法。逆向逐步代码重构法首先分析并建立嵌入式 Linux 与嵌入式 Windows 的程序接口对照表，然后根据对照表将源代码中与目标环境有差异的变量、程序结构、函数接口和数据库接口等内容进行转换，接着对修改后的源代码在目标平台上编译运行。逆向总体代码重构法则是从分析软件的总体结构入手，逆向设计软件的逻辑流程及各模块的功能与接口，基于目标平台实现软件的总体设计，接着在目标平台上采用特定编程技术对软件进行二次开发，完成软件系统的性能测试与功能优化[40,41]。

由于嵌入式硬件系统资源有限，在嵌入式操作系统中也只能采用精简的网络协议栈来支持网络操作。采用异类操作系统的目标板和宿主机之间、目标板和目标板之间，一般仍是采用(精简的)TCP/IP 协议栈作为网络互操作的基础，具体网络应用协议和具体网络应用程序编程接口(套接字)都是基于(精简的)TCP/IP 协议栈实现的。Windows Embedded CE 提供了两种功能强弱程度不同从而体积大小不同的轻量 TCP/IP 协议栈，以满足不同的嵌入式网络需求。在嵌入式 Linux 系统中采用的轻量 TCP/IP 协议栈，既可以来自对桌面 Linux 系统中 TCP/IP 协议栈的精简，也可以来自对开源的轻量级的与操作系统无关的 TCP/IP 协议栈(如 LwIP，LightWeight tcp/IP protocol stack)的移植[42]。以轻量 TCP/IP 协议栈为基础，嵌入式 Linux/Windows 目标板就可以支持具体网络应用协议和具体网络应用程序，与其它目标板或宿主机实现网络互操作来满足具体嵌入式应用的需求。

5 结语

综上所述，Linux 与 Windows 在桌面领域、网络领域和嵌入式领域都具有互操作性，并且两者的互操作技术还在不断地发展。在桌面领域，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。在网络领域，将 Linux 与 Windows 计算机集成在因特网中需

要开放、通用与统一的标准协议，CIFS 未来有望胜任这一协议，但还需要进一步完善和推广。在嵌入式领域，Linux 内核中的 KVM 技术未来有望在整合 Linux 与 Windows 嵌入式交叉开发平台中发挥重要作用。以 Linux 为代表的开源软件和以 Windows 为代表的商业软件之间在各个层面都存在互操作问题。开源软件在实现互操作方面具有天然的优势，而商业软件难以摆脱商业利益的羁绊。随着开源软件运动在全球的蓬勃发展，执行开放标准，解决互操作问题，已经成为软件产业的发展趋势。

参考文献

1 陆首群.探讨开源软件的互操作策略.软件世界,2007,(6):57–59.

2 潘爱民.Windows 内核原理与实现.北京:电子工业出版社,2010.

3 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.

4 NDISwrapper.http://wiki.debian.org/NdisWrapper,2010.

5 毛德操.关于 Longene 开发;漫谈兼容内核;漫谈 Wine.http://www.longene.org/whitebook.php, 2010.

6 Ward B. The Book of VMWARE. Sebastopol, California, USA: OREILLY & ASSOCIATES INC,2005.

7 Kernel–based Virtual Machine.http://en.wikipedia.org/wiki /Kernel–based_Virtual_Machine,2010.

8 Services for UNIX.http://www.microsoft.com/china/windowsserver2003/sfu/ default.mspx,2010.

9 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.

10 Linux NTFS file system support.http://sourceforge.net /projects /linux-ntfs/, 2010.

11 Ext2Fsd Project FAQ.http://www.ext2fsd.com/page_id=7,2010.

12 EXT2 IFS for Windows.http://uranus.chrysocome.net /linux /ext2ifs.htm, 2010.

13 Filesystem in Userspace.http://fuse.sourceforge.net/,2010.

14 ntfsprogs.http://en.wikipedia.org /wiki /Ntfsprogs# Included_programs,2010.

15 NTFS–3G Manual.http://www.tuxera.com/community/ntfs-3g–manual/,2010.

16 Dokan.http://dokan-dev.net/en/,2010.

17 Captive:The first free NTFS read/write filesystem for GNU/Linux.http://www.jankratochvil.net/project/captive/,2010.

18 Barrett DJ, Silverman RE.SSH 权威指南.冯锐,由渊霞(译).北京:中国电力出版社,2003.

19 SSH Filesystem.http://fuse.sourceforge.net/sshfs.html,2010.

20 Open source Dokan SSHFS.http://dokan-dev.net /en/2010/01/18/ open–source–dokan–sshfs/,2010.

21 Virtual Network Computing.http:// www.hep.phy.cam.ac.uk/vnc_docs/index.html,2010.

22 王悦.RDP 协议的安全性分析与中间人攻击.北京:北京邮电大学,2008.

23 Window X. System.http://en.wikipedia.org /wiki/X_Window_System,2010.

24 Independent Computing Architecture.http://en.wikipedia.org /wiki /Independent_ Computing_Architecture,2010.

25 安全套接层 SSL(Secure Sockets Layer).http://wenku.baidu.com /view /670da45177232f60dd cca161.html, 2010.

26 林海.浏览器/服务器应用开发.北京:科学出版社,2000.

27 Just what is SMB.http://samba.anu.edu.au /cifs/docs/ what–is–smb.html, 2010.

28 Hertel C, Hertel CR. Implementing CIFS.Upper Saddle River,New Jersey,USA:Prentice Hall PTR,2003.

29 Stern, Hal/Eisler, Mike/ Labiaga, Ricardo. Managing NFS and NIS.Sebastopol,CA,USA:Oreilly & Associates Inc, 2005.

30 Linux 网络文件系统.http://cooldatabase. Javaeye.com/blog/628085, 2010.

31 User Space NFS.http://unfs.sourceforge.net/,2010.

32 ProNFS–NFS client and server by Labtam.http://www.Pronfs.com/,2010.

33 iSCSI 存储技术全攻略.http://www.sansky.net/article /2007–12–03–iscsi –storage.html,2007.

34 Gay WW.实战 Linux Socket 编程.詹俊鹊,于卫(译). 西安:西安电子科技大学出版社,2002.

35 Wehrle K, Pahlke F.Linux 网络体系结构:Linux 内核中网络协议的设计与实现.汪青青,卢祖英译.北京:清华大学出版社,2006.

36 王大伟,杨凯锋,胡熠,等.用 Windows NT DDK 开发协议驱动程序.计算机与通信,1999,(8),64–67.

37 董平,刘心松.NT 下的协议驱动程序.计算机应用,2000,20(4),36–38.

38 田军营,韩建海.利用 Vmware 整合嵌入式系统开发环境.自动化与仪器仪表,2006,(5),86–88.

39 刘世栋,杨林,李京鹏等.Visual C++编制的 Windows 动态库移植到 Linux.微计算机应用,2003,24(3),165–169.

40 Fowler M. Refactoring: Improving the Design of Existing Code. Upper Saddle River, New Jersey, USA: Addison-Wesley Professional, 2005.

41 Code refactoring. http://en.wikipedia.org/wiki/Code_refactoring,2009.

42 Lwip–A Lightweight TCP/IP stack–Summary.http://savannah.nongnu.org /projects /lwip/,2010.