



中标软件有限公司

从Windows到Linux 的应用移植实现

——平台技术与接口篇

■ 兰雨晴 洪雪玉 著



国防工业出版社
National Defense Industry Press

从 Windows 到 Linux 的 应用移植实现

——平台技术与接口篇

兰雨晴 洪雪玉 著

国防工业出版社

·北京·

内 容 简 介

应用系统的开发、运行与部署依赖于所选择平台的工具、语言、环境、开发技术、服务及其兼容的第三方软件。因此进行移植时,需要考虑目标平台对上述因素的支持和实现情况。具体地,可以从应用系统组件、用户工作环境、开发工具/语言/平台、应用开发技术等几个角度整体考虑。本书主要关注应用开发技术在 Windows 平台和 Linux 平台的实现情况,研究分析网络通信、多进程/多线程、图形界面等技术在两个平台的实现差异性,并提供对应的实现方案。

在内容编排上,为了让读者对应用移植涉及的工作和过程有清晰的认识,本书在第一部分定义了应用移植过程、主要阶段、各阶段主要工作、应用移植的几个角度等,然后分为不同的部分,介绍多项应用开发技术的移植实现。其中,第二部分从第 2 章到第 5 章,介绍基于套接字网络通信技术的应用移植实现;第三部分从第 6 章到第 9 章,介绍基于并发多任务开发技术的应用移植实现;第四部分从第 10 章到第 12 章,介绍图形用户界面的应用移植实现。

图书在版编目(CIP)数据

从 Windows 到 Linux 的应用移植实现. 平台技术与接口篇/兰雨晴, 洪雪玉著. —北京: 国防工业出版社, 2013.1

ISBN 978-7-118-08252-4

I. ①从... II. ①兰... ②洪... III. ①操作系统-接口技术 IV. ①TP316

中国版本图书馆 CIP 数据核字 (2012) 第 206940 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 15 $\frac{1}{4}$ 字数 355 千字

2013 年 1 月第 1 版第 1 次印刷 印数 1—7000 册 定价: 38.00 元

(本书如有印装错误, 我社负责调换)

国防书店: (010)88540777

发行邮购: (010)88540776

发行传真: (010)88540755

发行业务: (010)88540717



前 言

随着计算机和网络技术的不断发展，信息化技术在各行各业获得了广泛而深入的应用。国防、政府、金融等关键领域借助信息化，不仅提高了工作效率，改善了业务流程，而且在信息的存储、共享、交互方面获得了更为高效和安全的实现手段。

然而，目前的信息化解决方案大都采用以 Windows 等国外商用操作系统为主的基础软件体制。操作系统之上的数据库管理系统、网络与通信软件、通用中间件、开发编译工具也以该商用操作系统之上的对应软件或者支持异构操作系统的软件为主。这就带来了整个软件体系的最根本问题，即过分依赖国外商用操作系统，系统的自主性和安全性难以得到保障，这对于国防、政府、金融等关键领域的应用系统尤为重要。

因此，迫切需要构建自有知识产权的基础软件体制，首当其冲的是研制国产操作系统。Linux 及其开源技术的出现，为我国研制国产操作系统提供了很好的契机和平台。借助 Linux 开源技术，在我国各级政府的大力支持下，国产 Linux 操作系统获得了前所未有的发展。以中标软件有限公司（以下简称“中标软件”）为代表的国产 Linux 企业在产品研发、应用开发、相关国内标准的出台等方面均获得了长足进步，积累了丰富的产品研发和应用经验，在金融、教育、财税、公安、审计、交通、医疗、制造等行业有大量成功应用，形成了一系列成熟的行业解决方案。

近年来，越来越多的用户已经把应用系统移植到国产软件平台上，或者正打算这么做。随着国产操作系统及其配套基础软件的进一步完善，基于国产基础软件构建我国关键领域应用的时机越发成熟。在国家“十一五”和“十二五”计划的推动下，预计未来几年，将有大量的应用系统需要移植到国产软件平台上。

由于 Windows 等国外商用系统与 Linux 系统在平台实现和开发支持方面具有较大差异，因此，实现从 Windows 到 Linux 系统的应用移植，需要解决二者的差异性对应用系统开发产生的影响。为了帮助用户顺利实现应用系统移植，中标软件专门成立了技术团队，专注于应用移植技术的研究及移植方案的设计实现。

应用系统的开发、运行与部署依赖于所选择平台的工具、语言、环境、开发技术、服务及其兼容的第三方软件。因此进行移植时，需要考虑目标平台对上述因素

的支持和实现情况。具体地，可以从应用系统组件、用户工作环境、开发工具/语言/平台、应用开发技术等几个角度整体考虑。本书主要关注应用开发技术在 Windows 平台和 Linux 平台的实现情况，研究分析网络通信、多进程/多线程、图形界面等技术在两个平台的实现差异性，并提供对应的实现方案。

在内容编排上，为了让读者对应用移植涉及的工作和过程有清晰的认识，本书在第一部分定义了应用移植过程、主要阶段、各阶段主要工作、应用移植的几个角度等，然后分为不同的部分，介绍多项应用开发技术的移植实现。其中，第二部分从第 2 章到第 5 章，介绍基于套接字网络通信技术的应用移植实现；第三部分从第 6 章到第 9 章，介绍基于并发多任务开发技术的应用移植实现；第四部分从第 10 章到第 12 章，介绍图形用户界面的应用移植实现。

本书可作为参考用书，主要针对专业技术人员，诸如软件分析人员、设计人员、开发者、软件工程师和编程人员等。如果需要了解应用移植过程、移植策略、主体工作、注意事项等，建议阅读第 1 章；如果需要从整体上大致了解 Windows 和 Linux 在上述应用开发技术方面的实现差异，建议阅读第 1 章和各个部分的首章节；如果需要了解两个平台在具体接口方面的差异性，建议阅读相关部分的所有章节。

本书是在中标软件承担的“核高基”重大专项课题支持下完成的（课题编号：2010ZX01036-001-001，2010ZX01037-001-001），由兰雨晴、洪雪玉著，同时感谢胡波、贾艳彬和刘勤为此书的形成提供了有力的技术支持；感谢余丹博士提供的宝贵意见。

由于时间仓促，书中难免存在错误和不足之处，欢迎读者批评指正。

著者于北京
2012.10

目 录

第一部分 基本理论

第 1 章 应用系统迁移移植基础.....	1
1.1 必要性分析.....	1
1.2 迁移移植原则.....	1
1.3 迁移移植过程.....	2
1.3.1 过程定义.....	2
1.3.2 迁移需求分析.....	2
1.3.3 关键技术迁移设计.....	3
1.3.4 编码迁移移植.....	4
1.4 平台差异性与移植对策.....	5
1.4.1 应用系统组件.....	5
1.4.2 用户工作环境.....	6
1.4.3 开发工具/语言/平台.....	7
1.4.4 应用开发技术.....	7
1.4.5 本书主要内容.....	8

第二部分 基于套接字网络通信技术的应用系统迁移移植

第 2 章 可移植性分析.....	9
2.1 平台实现.....	9
2.2 差异性分析.....	10
2.2.1 SOCKET 数据类型.....	10
2.2.2 fd_set 宏定义.....	10
2.2.3 错误码.....	11
2.2.4 资源限制.....	11
2.2.5 函数名称.....	11
第 3 章 基本功能迁移移植.....	13
3.1 创建和连接.....	13
3.1.1 接口差异性.....	13

3.1.2	移植实现.....	18
3.2	通信和关闭.....	19
3.2.1	接口差异性.....	20
3.2.2	移植实现.....	23
3.3	字节序.....	24
3.3.1	接口差异性.....	25
3.3.2	移植实现.....	25
第 4 章	高级功能迁移移植	27
4.1	socket 选项	27
4.1.1	接口差异性.....	27
4.1.2	移植实现.....	28
4.2	I/O 控制	30
4.2.1	接口差异性.....	30
4.2.2	移植实现.....	32
4.3	名称与地址转换.....	33
4.3.1	接口差异性.....	34
4.3.2	移植实现.....	39
第 5 章	I/O 模型迁移移植.....	41
5.1	模型概述.....	41
5.2	异步阻塞.....	42
5.2.1	实现概述.....	42
5.2.2	接口差异性.....	42
5.2.3	移植实现.....	43
5.3	异步选择.....	45
5.3.1	实现概述.....	45
5.3.2	接口差异性.....	46
5.3.3	移植实现.....	47
5.4	事件选择.....	54
5.4.1	实现概述.....	54
5.4.2	接口差异性.....	54
5.4.3	移植实现.....	55
5.5	重叠 I/O	62
5.5.1	实现概述.....	62
5.5.2	接口差异性.....	62
5.5.3	移植实现.....	64
5.6	完成端口.....	79
5.6.1	实现概述.....	79
5.6.2	接口差异性.....	80
5.6.3	移植实现.....	82

第三部分 基于并发多任务开发技术的应用系统迁移移植

第 6 章	可移植性分析.....	90
6.1	线程模型.....	90
6.2	差异性分析.....	91
6.2.1	进程创建.....	91
6.2.2	线程创建.....	92
6.2.3	IPC 通信机制	92
第 7 章	进程编程迁移移植.....	93
7.1	进程创建与退出.....	93
7.1.1	创建进程.....	93
7.1.2	进程退出.....	94
7.2	进程控制.....	95
7.2.1	终止进程.....	95
7.2.2	进程同步.....	95
7.3	进程属性.....	96
7.3.1	环境变量.....	97
7.3.2	进程优先级.....	98
7.3.3	当前进程 ID	98
7.4	移植实现.....	99
第 8 章	线程编程迁移移植.....	101
8.1	线程创建与退出.....	101
8.1.1	创建线程.....	101
8.1.2	线程退出.....	102
8.2	线程控制.....	103
8.2.1	终止线程.....	103
8.2.2	挂起/恢复线程	104
8.2.3	线程同步.....	105
8.3	线程属性.....	105
8.3.1	线程优先级.....	106
8.3.2	线程 ID	106
8.4	线程本地存储.....	107
8.4.1	分配与释放索引.....	107
8.4.2	获取与设置数据.....	108
8.5	移植实现.....	109
第 9 章	IPC 机制迁移移植	112
9.1	匿名管道.....	112

9.1.1	创建与关闭.....	113
9.1.2	读写操作.....	113
9.1.3	移植实现.....	114
9.2	内存映射文件.....	118
9.2.1	创建与关闭.....	119
9.2.2	连接与分离.....	121
9.2.3	移植实现.....	122
9.3	邮箱.....	126
9.3.1	创建与关闭.....	127
9.3.2	读写操作.....	128
9.3.3	移植实现.....	129
9.4	互斥锁.....	131
9.4.1	创建与销毁.....	132
9.4.2	加锁与解锁.....	133
9.4.3	实现.....	133
9.5	信号量.....	136
9.5.1	创建与销毁.....	137
9.5.2	获取与释放.....	139
9.5.3	移植实现.....	140
9.6	事件.....	142
9.6.1	创建与销毁.....	144
9.6.2	等待事件.....	145
9.6.3	设置与重置.....	145
9.6.4	移植实现.....	146
9.7	临界区.....	149
9.7.1	初始化与销毁.....	150
9.7.2	进入与离开.....	151
9.7.3	移植实现.....	151

第四部分 图形用户界面迁移移植

第 10 章	应用开发框架.....	155
10.1	框架模式.....	155
10.1.1	MFC 文档/视图.....	155
10.1.2	Qt 模型/视图.....	157
10.1.3	相关开发类.....	159
10.2	应用类型.....	162
10.3	消息响应机制.....	165

10.3.1	MFC 消息机制	165
10.3.2	Qt 信号槽机制	167
第 11 章	图形绘制基础设施	168
11.1	绘图系统	168
11.2	核心能力	168
11.2.1	图元绘制	169
11.2.2	位图处理	169
11.2.3	图元文件	171
11.2.4	打印操作	171
11.2.5	路径绘制	172
11.2.6	属性设置	173
11.2.7	坐标及映射	174
11.3	特效支持	176
11.3.1	反走样技术	176
11.3.2	半透明效果	177
11.4	绘图工具类	180
11.4.1	画笔	180
11.4.2	画刷	182
11.4.3	字体	184
11.4.4	调色板	186
11.4.5	区域	188
11.5	移植实例	189
11.5.1	Windows 实现	189
11.5.2	移植分析	189
11.5.3	Linux 对应实现	191
第 12 章	图形界面开发类	194
12.1	开发类结构	194
12.2	对话框类	196
12.2.1	字体选择对话框	196
12.2.2	颜色选择对话框	198
12.2.3	文件选择对话框	200
12.3	控件类	203
12.3.1	按钮控件	204
12.3.2	编辑框控件	206
12.3.3	组合框	209
12.3.4	列表框控件	212
12.3.5	进度条控件	214
12.3.6	列表视图控件	216
12.3.7	树形视图控件	220

12.3.8 滚动条	224
12.3.9 工具条	226
12.3.10 选项卡控件	228
12.4 布局类	231
12.5 移植实现	232
12.5.1 Windows 实现	232
12.5.2 迁移移植分析	233
12.5.3 Linux 对应实现	234
参考文献	240

第一部分 基本理论

第 1 章 应用系统迁移移植基础

1.1 必要性分析

目前,包括国防、政府、金融等关键领域在内的行业应用主要采用以国外商用操作系统为主的基础软件体制。操作系统之上的数据库管理系统、网络与通信软件、通用中间件、开发编译工具也以该商用操作系统之上的对应软件或者支持异构操作系统的软件为主。这就带来了整个软件体系的最根本问题,即过分依赖国外商用操作系统,系统的自主性和安全性难以得到保障。而采用国产操作系统可有效规避上述问题。考虑到当前大量的行业应用已经在国外商用操作系统下开发完成并得到应用,完全重新开发不仅费时费力,浪费大量的资源,甚至有可能损失已投入或积累的数字资产。因此,考虑将已有应用向国产操作系统上迁移具有很大的现实意义。

但迁移实施时,需要考虑和解决一系列相关问题,例如,怎么开展迁移移植,实施过程如何?源平台和目标平台在开发技术方面的差异性?目标平台的对应技术方案?源平台和目标平台对常用编程语言和工具支持的差异性?目标平台的对应开发环境?如何保证原应用系统的软硬件投资?如何保证与其他应用系统的互联互通?等等。我们将专注于解决上述问题,保证应用系统的顺利迁移。重点关注源平台和目标平台的技术差异性,并制定相应的迁移移植方案。

1.2 迁移移植原则

随着越来越多的企业和用户认可 Linux 的价值,人才问题也日益凸显,成为阻碍 Linux 在我国快速发展的瓶颈之一。Linux 专业人才,尤其是本地化专业人才,无论从数量还是质量,与 Windows 人才相比存在很大差距。随着企业对 Linux 人才需求的不断增加,对人才结构和层次也有更高要求,包括对系统级的数据库、消息管理、Web 应用、桌面办公、各种嵌入式开发等方面的人才需求。另一方面,对普通用户而言,对 Linux 系统大多缺乏了解,不熟悉其操作和使用,至于 Linux 应用开发经验更是匮乏。

鉴于目前我国 Linux 人才现状,完成从 Windows 平台到 Linux 平台的应用系统迁移移植,对用户而言将是一个巨大的挑战,他们需要解决以下问题:

- (1) 与 Windows 商用平台相比, Linux 平台的可用性如何?
- (2) 对各类应用系统的功能及性能需求, Linux 是否具备成熟的开发技术?

- (3) Linux 是否易于开发和使用？是否具备成熟完备的开发工具和集成开发环境？
- (4) 原有的软件资源、硬件资源是否可重用？
- (5) 应用系统移植后，如何保持与其他应用系统的互联互通？

由此可见，实施应用系统迁移移植时，用户在项目前期需要投入较多的学习成本，掌握 Linux 系统的基本开发技术。由于对 Linux 项目经验和成熟技术方案的缺乏，也导致了一定的技术风险。可能影响项目实施时间和成本，导致其不可控，甚至导致项目实施失败。

中标软件作为我国主要的 Linux 操作系统提供商之一，除了研发满足用户需求的高质量产品，有责任协助用户完成各类应用系统的迁移移植，推动 Linux 系统在国内的普及应用。为了弥补用户经验的不足，帮助其顺利完成迁移移植工作，迁移实施时我们将遵循以下原则：

- (1) 帮助用户减少开发时间和学习成本；
- (2) 帮助降低项目迁移移植的技术风险；
- (3) 尽量采用标准化、跨平台开发技术；
- (4) 最大限度保护用户原有投资，重用已投入使用的硬件设备、第三方开发库和工具等。

1.3 迁移移植过程

1.3.1 过程定义

随着“十二五”相关工作的开展，将会有越来越多的行业应用，例如政府、金融、教育、交通、农业等，需要迁移到 Linux 平台下，因而有关迁移移植技术的研究就显得尤为重要。作为专业的 Linux 厂商，中标软件应对 Linux 发展需求和应用系统迁移需求，已经开始了针对相关迁移移植技术的探索步伐。

迁移移植需要考虑和解决一系列的技术问题，例如，Windows 平台和 Linux 平台在开发能力和技术方面的差异性。由于应用系统类型多样，依赖的平台技术可能存在差异，迁移移植的复杂度和工作量也各不相同。为了保障迁移工作的顺利开展，首先有必要确立一个项目实施过程，以统一对整个过程中主要工作的认识。

一般地，研发一个应用系统，需要经历“需求—设计—编码—测试—部署”等阶段。根据项目迁移移植经验，我们认为，实施应用系统迁移移植也需要经历类似的阶段，即“迁移需求分析—关键技术迁移设计—编码迁移移植—测试—部署”，如图 1-1 所示。其中，前三个阶段最为关键，是迁移移植的主体工作。

迁移移植涉及两个层面：设计级迁移移植和编码级迁移移植，分别对应整个实施过程的第二阶段和第三阶段。前者需要针对各项关键技术，在目标平台设计相应的实现方案；而后者在确定技术方案后，基于目标平台开发工具，采用具体的编程接口替换原有接口，或重写部分代码。

1.3.2 迁移需求分析

迁移移植过程的第一阶段是迁移需求分析。此阶段需要全面了解原应用系统的各个方面，可能包括：

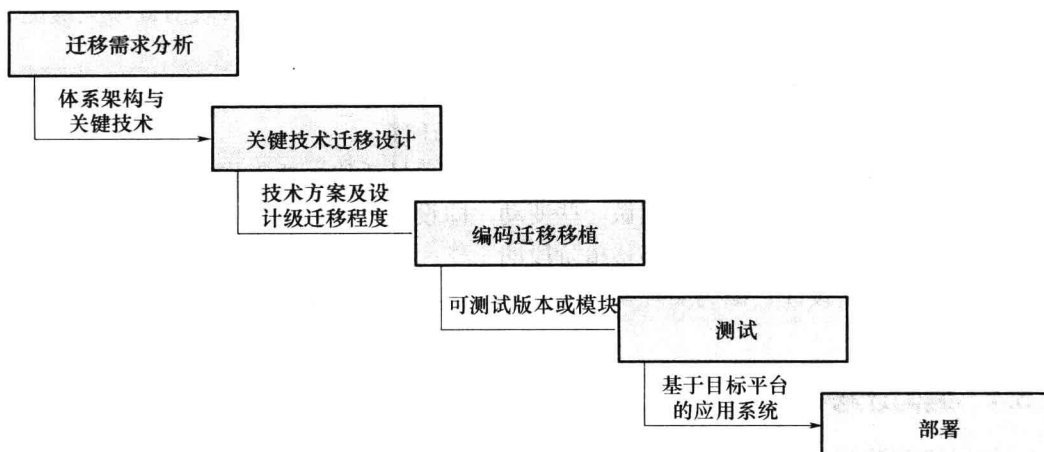


图 1-1 迁移移植过程及各阶段关系

- (1) 应用系统目标：软件用途、应用场景、服务对象等；
- (2) 应用系统结构：软件模型、数据库、中间件、其他第三方软件、硬件平台及主要外设等；
- (3) 功能与性能：主体功能、网络性能、I/O 性能、可靠性、安全性等；
- (4) 应用开发技术：技术框架、关键技术及实现方案、开发平台及语言工具；
- (5) 其他：实际环境约束与限制、用户知识情况、系统部署等。

1.3.3 关键技术迁移设计

关键技术迁移设计是迁移实施过程的第二阶段。此阶段是整个迁移移植成败的关键所在，需要深入分析原应用系统各项关键能力的具体需求，并在目标平台上设计相应的技术方案，此时有几种可能（见图 1-2）：

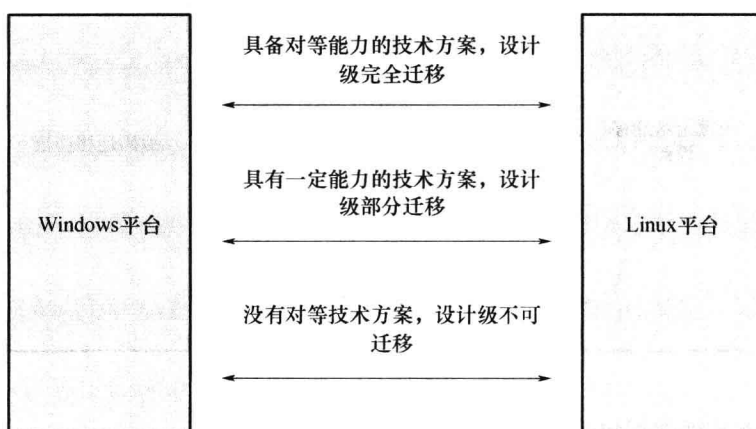


图 1-2 关键技术迁移程度

- (1) 目标平台有对等能力的技术方案，此项关键能力可以平滑地迁移到目标平台，称之为设计级完全迁移；

(2) 目标平台没有完全对等的技术方案，但可以通过优化手段实现对此项关键能力的支持，称之为设计级部分迁移；

(3) 原开发技术为原系统平台专有技术，如.NET 技术，此时需要在目标平台上重新设计和编码实现整个应用系统，称之为设计级不可迁移。

如果设计级完全迁移，无需修改原设计方案，项目迁移主要工作量体现在编码迁移移植阶段；如果需要对原设计方案做一些变动，即设计级部分迁移，项目迁移主要工作量体现在部分设计优化和编码迁移移植阶段的工作；如果设计级不可迁移，需要基于目标平台能力，重新设计、编码实现整个应用软件，其工作量最大，相当于重新开发一个应用系统。

1.3.4 编码迁移移植

完成了关键技术迁移设计，进入编码迁移移植阶段。此阶段的主要工作是，基于迁移技术方案，借助目标平台开发工具，采用具体的编程接口实现。这阶段原则上没有技术风险，工作量由两个方面决定；上一阶段的结果以及所采用的编程语言。

1. 根据关键技术迁移设计阶段的结果

如果设计级完全迁移，则主要工作为基于目标平台的开发接口编码实现或做接口替换；如果设计级部分迁移，则主要工作包括两部分，一部分与设计级完全迁移时的工作类似，另一部分为用于支持技术方案的代码优化工作；如果设计级不可迁移，则需要在目标平台重新设计、编码实现整个软件系统。

由此可见，随着设计方案迁移程度从高到底，编码迁移移植阶段的工作量也随之增加（见图 1-3）。

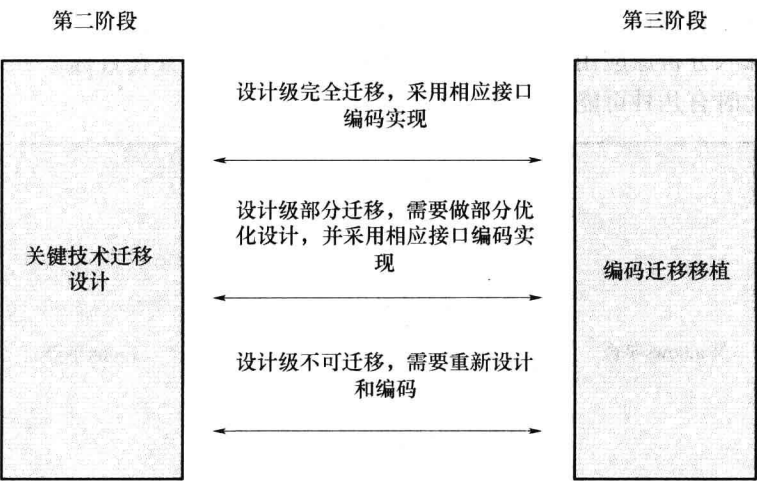


图 1-3 编码迁移移植主体工作

2. 根据所采用编程语言

目前主流编程语言大致分为两类：平台通用语言和平台专用语言。前者如 C、C++、Java、PHP、Perl、Python 等，后者如 C#、VB、Delphi 等。原则上讲，基于通用语言的迁移移植，其工作量较小，而基于专用语言的迁移工作量较大，具体分析如下。

基于 Java 语言开发的应用系统，Linux 系统提供了符合工业标准的 Java 虚拟机，凭借其跨平台特点，应用系统迁移时，一般无需修改代码，就可以在目标平台上运行了，具有移植快、工作量小的优点。

对于 C/C++语言开发的应用系统，Linux 系统将从两个方面提供迁移移植支持。首先，提供一个功能强大、高性能、高效率的 C/C++ 编译器，并支持应用程序的优化编译；第二，提供相应功能的 C/C++开发库支持，如输入/输出、内存管理、字符串处理等 C 语言开发库，以及 Qt 等 C++开发库。由于 Windows 和 Linux 系统提供的 C/C++编译器均大致兼容工业 C/C++语言编译器标准，且均提供功能相当的开发库，因此这类应用程序迁移移植的工作量也不大，主要工作是采用对等的编程接口实现。

如果原应用系统采用 C#、VB、Delphi 等 Windows 平台专用语言实现，迁移移植时，需要选择目标平台支持的语言，重新设计、编码实现整个软件系统。因此这类应用系统的迁移移植工作量最大。

1.4 平台差异性与移植对策

应用系统的开发、运行与部署依赖于所选择平台的工具、语言、环境、开发技术、服务，及其兼容的第三方软件。因此实施迁移移植时，需要考虑目标平台对上述因素的支持和实现情况。具体地，可以从应用系统组件、用户工作环境、开发工具/语言/平台、应用开发技术等几个角度整体考虑。

1.4.1 应用系统组件

应用系统类型多样，如企业信息管理系统、协同办公系统、电子政务、电子商务、数据共享平台等。为了支撑其业务功能，通常需要基于数据库、中间件或一系列基础服务来构建，如图 1-4 所示。

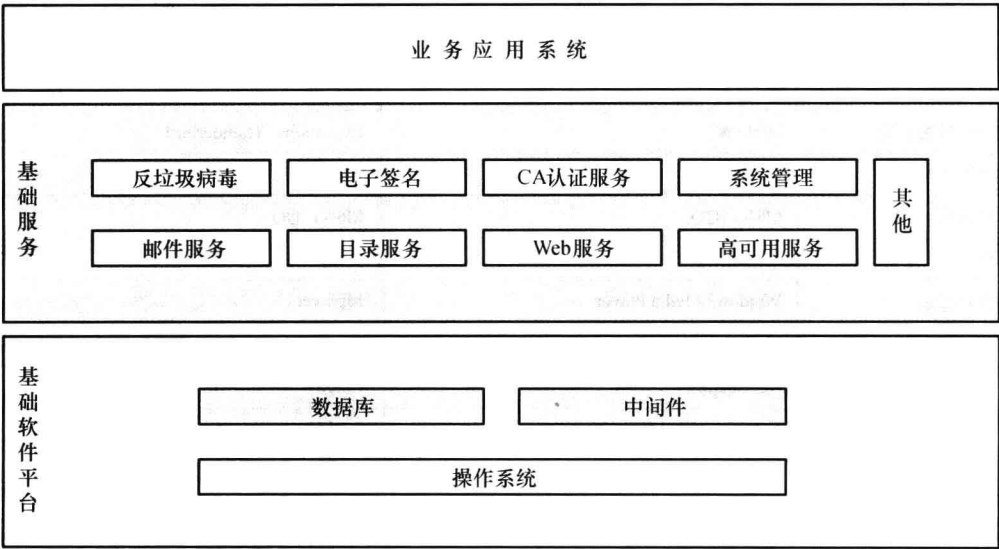


图 1-4 应用系统组件

Linux 系统对主流数据库和中间件均提供很好的支持；支持文件服务、共享服务、打印服务、电子签名、邮件服务、目录服务、Web 服务等基础应用服务。表 1-1 列举了部分软件组件及其对应的 Windows 和 Linux 支持情况。

表 1-1 Windows 和 Linux 下部分常用的基础软件和应用服务

系统组件	Windows	Linux
Web 服务	IIS	Apache
FTP 服务	SerU, FTP server	Vsftpd, wu-ftp
文件共享	Windows Share	Samba, NFS
邮件服务	MDaemon, Hamster	Sendmail, Postfix
群件服务	Microsoft Exchange	CommuniGate Pro
目录服务	Active Directory	OpenLDAP
数据库	MS SQL, MySQL, IBM DB2, Oracle, Informix	PostgreSQL, MySQL, IBM DB2, Oracle, Informix
中间件	Weblogic, WebSphere, Tomcat	Weblogic, WebSphere, Tomcat, JBoss

应用系统迁移时，如果涉及上述组件，可以采用 Linux 系统支持的对应组件进行替换。移植时的主要工作可能包括：目标组件的安装部署和配置、原有文件和数据库等信息资源的移植、脚本代码的移植等。

1.4.2 用户工作环境

Windows 下有大量的应用程序和工具，Linux 下也有对应或类似的实现，完全可以满足用户日常工作和娱乐需求。表 1-2 列举了部分常用的软件工具及其对应的 Windows 和 Linux 实现。

表 1-2 Windows 和 Linux 下部分常用的软件工具

软件工具	Windows	Linux
办公软件	Microsoft Office	中标普华 Office, Open Office
网页浏览	IE	Firefox
电子邮件客户端	Outlook	Evolution, Thunderbird
FTP 客户端	CuteFTP	filezilla
即时通信	MSN, QQ	MSN, QQ
图像处理	PhotoShop	Gimp
多媒体播放	Windows Media Player	Mplayer
文件下载	Flashget	Wget
音视频会议	NetMeeting	Ekiga
防火墙	Norton Internet Security	iptables
PDF 阅读器	Adobe Acrobat Distiller	Adobe Reader, Xpdf
项目管理	Microsoft Project Manager	Planner
流程图绘制	Visio	Kivio, Dia