

Novell's Guide to
NetWare 4.0 NLM Programming
Michael Day Michael Koontz Daniel Marshall

Novell指南 NetWare 4.0 NLM 程序设计

李卫国 张秉南 姜圣国 译
李卫国校



電子工業出版社

NetWare 4.0 NLM 程序设计

[美] Michael Day Michael Koontz Daniel Marshall 著

李卫国 张秉南 姜圣国 译

李卫国 校

电子工业出版社

(京)新登字 055 号

内 容 提 要

本书内容包括 NLM 程序员所需的基本知识,写多线程 NLMs 程序,目录服务,基本和高级文件系统程序设计,IPX/SPX 同 SPX II 通信,用扩展 NCP 编写分布式应用程序,存储管理和保护,编写与多种 NetWare 版本兼容的 NLM,NetWare 的移植,名字空间、数据流和扩充属性。

本书是为正在进行 NLM 开发的软件工程师、程序设计员以及开发管理人员编写的。

Novell's Guide to NetWare 4.0 NLM Programming Copyright(c) 1993 by Michael Day Michael Koontz Daniel Marshall. Chinese translation Copyright(c) 1994 by Publishing House of Electronics Industry.

中文简体字版专有版权(c)1994 电子工业出版社。

Publishing by arrangement with Novell Press. Copyright licensed by Cribb-Wang-Chen, Inc. /Bardon-Chinese Media Agency.

本书经博达著作权代理有限公司安排取得。

All Right Reserved.

NOVELL 指南

NetWare 4.0 NLM 程序设计

[美] Michael Day Michael Koontz Daniel Marshall 著

李卫国 张秉南 姜圣国 译

李卫国 校

责任编辑 陆伯雄

*

电子工业出版社出版(北京市万寿路)
电子工业出版社发行 各地新华书店经销
北京市顺义县天竺颖华印刷厂印刷

*

开本: 787×1092 毫米 1/16 印张: 25.875 字数: 650 千字

1994 年 4 月第 1 版 1994 年 4 月第 1 次印刷

印数: 6000 册 定价: 52.00 元

ISBN7 - 5053 - 2271 - 0/TP • 623

引 言

Novell 的 NetWare 4.0 NLM 程序设计指南既是一本参考手册也是 NLM(NetWare Loadable Module)软件开发指南。它包含了成千上万行源代码和一些非常有用的 NLM 应用范例,说明了在实际应用时怎样避免 NLM 开发过程中潜在的、易犯的错误。

本书是由工作在 NetWare 4.0 开发项目方面的工程技术人员编写的,许多 Novell 的工程师作为技术顾问参与了这项工作。因此,本书包含的信息对任何关注 NLM 开发的人都是非常有用的。

▶ ◀

本书是为谁编写的

本书是为打算或正在进行 NLM 开发的软件工程师、程序设计员以及开发管理人员编写的。如果你正好留心于 NetWare 4.0 的开发环境,本书可以回答你很多重要问题。

本书提供的信息对高级软件工程师及初级程序员都是一样有用的。几乎所有的例子都是用易读的 C 语言编写的。

本书包括的基本题目是:

- ▶ 链接和装入 NLMs
- ▶ 基本文件系统程序设计
- ▶ 编写 NLM 可移植性 C 代码
- ▶ NetWare 调试程序

本书包括的高级题目是:

- ▶ NetWare 目录服务程序设计
- ▶ 网络通信,包括 IPX/SPX, TCP/IP 和运输层接口(TLI)
- ▶ 多线程程序设计
- ▶ 分布式应用设计和开发
- ▶ 扩展的 NetWare 核心协议(NCPs)
- ▶ 异步文件系统程序设计
- ▶ 钩联 NetWare 文件系统
- ▶ 名字空间程序设计
- ▶ C 语言和 32 位汇编语言的结合

本书也包含了对不同的 NetWare 版本开发 NLMs 的有用信息,包括在 NetWare 交叉平台兼容性方面的专门章节,在这些平台上可开发出运行在 NetWare 3.11、NetWare 4.0 服务器和 NetWare SFT Ⅲ 3.11 上的 NLMs。

如果你对以上的任何题目感兴趣或简单地想获得网络程序设计技巧,本书是一个良好的选择,NLMs 作为网络应用的一个高级平台越来越得到了认可,并且 NLM 环境正在很快的成为主流开发平台。

▶ ◀

有关 NetWare 3. x 的内容

除介绍 NetWare 目录服务的第三章外,每章都提供了可用于 NetWare 3.11 的 NLMs 程序代码。所有在本书出现的程序设计技术都可直接应用于 NetWare 3. x 的 NLMs 程序设计。事实上,本书的全部章节都致力于交叉平台兼容和向下兼容。

▶ ◀

怎样从 Novell 得到范例程序代码

本书的示范程序代码通过计算服务中心可以得到,当你向计算中心注册后,输入:

```
GQ NOVLIB
GO LIBRARY 2
```

然后下载文件 NLM40PG. ZIP。

为得到问题的答案、报告故障或获得支持,可发送电子邮件到 Michael Day,计算中心地址是 {71670,475}。

▶ ◀

本书的结构

本书包括 11 章。

第一章:NLM 基础

第一章提供了所有 NLM 开发者所需的基本知识。在这一章你将了解到 NetWare 的 C 接口,它包括 NetWare 4.0 在 Intel 保护模式和实模式中的结构,还可以了解到怎样链接和装入 NLM,除此之外本章还介绍了 NetWare 4.0 其它方面的内容,如它的调度程序、多线程 (multithread) 执行模式和内存保护系统。最后,本章综述了 NetWare 4.0 所提供的的基本服务,

• X •

包括 NetWare 目录服务、装订库、IPX/SPX、流、NetWare 文件系统和 NetWare 屏幕处理系统。

第二章:写多线程 NLMs 程序

了解多线程的运行过程也许是程序员面临 NLMs 的新的最大的问题。NetWare 提供了独特的多线程结构,在 NLM 中使用多线程技术在性能和效率方面为程序员打开了新的视野,而在其它开发平台上不可能有这种效果。

在第二章中介绍的多线程程序设计技术实际上是很简单的,所有程序员都是可以掌握的,为了说明多线程程序设计技术,本章介绍了一个实际的多线程客户/服务器数据库程序的源代码,还可以了解到 NetWare 中用于建立和调度线程的不同方法,更重要的是,你还可以了解到 NetWare 4.0 的新的、更有效的调度程序以及怎样运用它。

第三章:目录服务

也许 NETWARE 4.0 的最重要特征是 NetWare 目录服务(NDS),它是按照 OSI 的分布式全球命名系统实现的。本章介绍了怎样在你的 NLMs 应用中使用如此丰富的 NDS API。

运用 NDS API 的 NLMs 应用可以获得极大的益处,使其具有真正的全球范围和广域网属性。然而,使用 NDS API 需要专门的、新的 NetWare 程序设计技术。第三章详细介绍了 NDS 程序设计技术,它为 NLM 应用打开了一扇通向世界前沿的成功之门。

第四章:基本文件系统程序设计

NetWare 4.0 支持工业标准的文件 I/O APIs,包括 ANSI C 的文件 I/O 库和 POSIX 的低级文件 I/O 子程序。而且,NetWare 文件系统提供的适于 NLM 程序员使用的非常基本的编码技术是更丰富和强有力的。

第四章集中于基本文件系统程序设计技术,该技术允许 NLMs 真正利用非对等 NetWare 文件系统。本章包括了一些非常有用的 NLM 实用程序的源代码,还包括了所有 NetWare 用户都熟悉的 NDIR、FLAG 和 DSPACE 实用程序的 NLM 版本。

第五章:高级文件系统程序设计

想更深入了解 NLMs 的程序员,可以阅读第五章。对于怎样使用 NetWare 高性能文件 I/O APIs 及 NetWare 异步文件系统,本章里有一些更有帮助的应用。它包括一个使用新版本文件系统监控接口的 NLM 应用,这个接口允许 NLMs 链接到 NetWare 文件系统。

第六章:IPX/SPX 同 SPXII 通信

开发好网络软件的关键是控制网络通信。NetWare 4.0 提供了一套丰富的网络通信 APIs,包括典型的 IPX/SPX 接口,增强的通用运输层接口(TLI)。使用 TLI,开发者可以使网络应用系统运行在 IPX/SPX 和 TCP/IP 之上,而不用修改应用程序代码。第六章包含一些简单的范例程序,它们介绍了在 NLMs 网络通信中不同的编程方式。

第七章:用扩展 NCP 编写分布式应用程序

开发分布式客户/服务器应用是现代科技发展的一个动态,它表明了当前和将来的计算所需要的进取方向。第七章包含一个分布式客户/服务器程序,并且列出了开发分布式应用所需要的步骤,还详述了 NCP 扩展接口。

第八章:存储管理和协议

NetWare 4.0 以可选的存储保护为特色,在这以前的 NetWare 缺乏这方面的内容。NetWare 4.0 存储保护接口则以对 NLM 开发者提供特别的选项为特点,它可以加速应用开发和调试过程。第八章提供的信息可以帮助你更好地利用 NetWare 4.0 存储保护的命令和设计要点。

第九章:编写与多种 NetWare 版本兼容的 NLM 程序

随着 NetWare 在世界范围的推广,开发者增加了建立交叉平台兼容的 NetWare 应用方面的要求。当前 32 位 NetWare 平台包括 NetWare 3.x、NetWare 4.x、NetWare for UNIX 以及 SFT Ⅲ NetWare。Novell 还为非 Intel 计算机开发了独立平台的 NetWare。

第九章详述了开发者保证兼容性 NLM 交叉平台最高质量的步骤。本章还包括一个可以装入和运行在 NetWare 3.x 和 NetWare 4.x 服务器上的 NLM 程序,这个 NLM 说明怎样感知 NetWare 目录服务的存在及怎样使用它。还说明如果 NDS 不存在时怎样使用装订库 APIs。第九章还包括 NetWare SFT Ⅲ 的专门内容,详细的步骤可以使开发者在镜像服务器上运行应用系统时获得好处。

第十章:NetWare 的移植

本章叙述了 Novell 工程师用于保证最可能的代码移植性的技术。包括在本章的内容是 Novell 在 NetWare 开发时使用的专门的头标文件和宏调用。

第十一章:名字空间、数据流和扩充属性

本章包含的名字空间程序设计内容,允许 NLMs 使用各种可能的文件格式。例如 UNIX、Macintosh 和 OS/2 HPFS 的文件格式。NetWare 4.0 具有一个比以前版本更开放的文件系统的特点,这对于在 NetWare 程序设计方面没什么了解的人将是格外重要的。

目 录

引 言	(IV)
第一章 NLM 基础	(1)
熟悉 NetWare 操作系统环境	(1)
了解 NLM 是什么?	(2)
NLM 能做什么?	(3)
怎样编译一个 NLM?	(3)
NLM 程序设计所需内容	(3)
熟悉 NetWare NLM C 接口库函数	(3)
怎样使你的 NLM 程序设计又快又好?	(4)
为什么要了解 CLIB 接口?	(4)
什么是 CLIB 中一般类型的函数?	(4)
怎样使用 NLM 库 APIs?	(6)
链接/装入程序	(7)
什么是 PRELUDE.OBJ?	(7)
NLM 链接程序的差异	(7)
怎样使用 NLM 链接程序?	(7)
了解 NetWare 核	(8)
核是怎样运行的?	(8)
内存管理	(8)
NetWare 是怎样使用内存的?	(8)
NetWare 提供怎样的内部保护机制以阻止 NLMs 相互间的破坏?	(9)
怎样分配和使用内存?	(9)
多线程程序设计	(9)
什么是多线程程序设计?	(9)
什么是多线程程序设计的基本内容?	(9)
NLM 上下文变量	(10)
什么是 NLM 上下文?	(10)
怎样建立 NLM 上下文?	(10)
屏幕处理系统	(10)
什么是屏幕?	(10)
推荐使用屏幕的方法	(11)
NLM 库	(11)
什么是 NLM 库?	(11)
怎样建立 NLM 库?	(11)

怎样使用 NLM 库?	(11)
目录服务和装订库	(12)
什么是目录服务?	(12)
DIB 和装订库之间的区别	(12)
什么是装订库仿真?	(12)
怎样访问目录?	(13)
了解 NetWare 4.0 文件系统	(13)
文件系统怎样执行?	(13)
名字空间	(13)
什么是名字空间?	(13)
怎样使用名字空间?	(14)
熟悉 SPX I	(14)
什么是 SPX I?	(14)
怎样区别 SPX I 与 IPX/IP?	(14)
怎样使用 SPX I?	(14)
熟悉 STREAMS	(15)
什么是 STREAMS?	(15)
为什么需要了解 STREAMS?	(15)
怎样使用 STREAMS?	(15)
小结	(15)
第二章 写多线程 NLMs 程序	(17)
线程和非优先级环境的定义	(17)
单线程的 NLMs	(17)
多线程的 NLMs	(18)
非优先级环境	(18)
NetWare OS 中的队列	(18)
线程与作业的比较:它们分别在 OS 中的工作原理	(19)
WorkToDoList 队列	(19)
RunList 队列	(19)
DelayedWorkToDoList 队列	(19)
LowPriorityRunList 队列	(20)
本章讨论的 C 接口子程序	(20)
一个多线程 NLM 例子	(23)
服务器一侧	(23)
客户方一侧	(24)
ENGINE NLM 的主线程	(24)
主程序为开始附加线程作准备	(27)
OPENSEMAPHORE 子程序	(29)
INITMAIN:缺省线程组中的附加线程	(29)

初始化监听、发送 ECB	(31)
为监听询问设置监听、发送 ECB	(32)
让 InitMain 睡眠	(32)
InitMain 从睡眠到唤醒	(34)
InitMain 建立线程组	(35)
EngineMain 填写监听 ECB(包括 SemHandle 字段)并且设置它们	(35)
EngineMain 睡眠	(36)
EngineMain 唤醒并得到呼入请求	(37)
EngineMain 处理 ECB 请求	(40)
使用 BeginThread 的选择方法	(44)
正常地退出 engine NLM	(44)
小结	(47)
第三章 NetWare 目录服务	(48)
目录服务的概念	(48)
定义目录信息库	(49)
了解目录纲要	(49)
解释什么是分区	(52)
回答访问问题	(52)
进一步考查目录服务的安全性	(53)
开始用目录服务函数	(58)
成为正式的 NetWare 4.0 客户	(60)
初始化目录上下文	(60)
通过注册建立凭证	(63)
进行证明	(63)
为目录信息库设置访问安全	(65)
目录服务的访问控制	(65)
改变 ACL 属性	(66)
安全等效性和组成员	(68)
改变目标的密钥对	(69)
查询目录树	(70)
通过浏览树来找出目标	(70)
执行黄页查询	(72)
两种检查属性值的方法	(74)
目录纲要函数	(76)
修改目录纲要	(77)
分区的目录树	(79)
一个简单的 NLM 应用	(81)
学习使用 DSSCRIPT	(81)
介绍 DSSCRIPT	(82)

练习运行 DSSCRIPT	(82)
学习怎样建立一个脚本文件	(83)
小结	(93)
第四章 基本文件系统程序设计	(94)
NetWare 4.0 文件系统概述	(94)
文件系统的向下兼容性	(94)
NetWare 卷	(94)
文件系统高速缓存	(96)
并发控制	(97)
事务控制	(97)
名字空间	(98)
稀疏文件	(98)
压缩	(98)
子地址分配	(99)
标准文件 I/O	(99)
低级标准 I/O	(99)
高级标准 I/O	(99)
对于 NetWare 的 DIR 命令	(101)
合法的 NetWare 路径	(110)
查询目录	(111)
打印文件的时间和日期信息	(112)
从目标 IDs 获得目标名	(113)
获得子目录信息	(114)
修改 DIR. NLM	(114)
文件属性处理	(115)
FLAG. NLM 的内容	(115)
处理命令行参数	(122)
处理文件属性	(123)
修改 FLAG. NLM	(124)
用户信息和控制	(124)
空间消耗控制	(124)
托管控制	(126)
修改继承权屏蔽	(129)
DSPACE. NLM	(129)
对 DSPACE. NLM 的修改	(138)
小结	(138)
第五章 高级文件系统编程	(139)
高性能文件 I/O	(139)
异步文件 I/O 注释	(141)

用 FEsopen 获得更多的控制	(141)
转移服务器注释	(143)
用共享的文件工作	(143)
标识文件共享	(143)
sopen 和 FEsopen	(144)
在一个可共享文件上加锁	(144)
对单个文件加锁和解锁	(145)
对一组文件加锁和解锁	(146)
加锁和释放单个记录	(147)
加锁和释放一组记录	(149)
一些文件或记录解锁,但不清除它们	(150)
逻辑锁与物理锁	(151)
NetWare 事务跟踪系统	(151)
什么是一个事务?	(151)
什么是一不完整事务?	(151)
隐含事务和明显事务	(152)
在你的 NLM 中使用 TTS	(152)
异步文件系统	(155)
异步地读文件数据	(155)
释放由 AsyncRead 所分配的存储区	(157)
异步地写文件数据	(158)
FILEIO. NLM	(159)
文件监视器接口	(170)
FMON. NLM	(171)
对 FMON. NLM 源程序的讨论	(191)
小结	(196)
第六章 IPX/SPX 与 SPX I 通信	(197)
IPX 编程	(197)
IPX 的组成部分	(198)
事件服务子程序(ESRS)的使用	(200)
Client/Server 应用程序的编程	(200)
SPX 的使用	(205)
IPX 与 SPX 的比较	(205)
把采用 IPX 的应用程序移植到 SPX	(205)
SPX I	(212)
SPX I 的评价	(213)
介绍用于 SPX I 中的 TLI	(213)
TLI 和 IPX	(216)
TLI 和 SPX	(220)

TLI 和 TCP	(226)
小结	(227)
第七章 用 NCP 扩展器编写分布式应用程序	(229)
NCP 扩展器如何工作	(229)
NCP 扩展器的 APIs	(230)
C 接口服务器 APIs	(230)
C 接口客户 APIs	(230)
NCP 扩展器的优点和缺点	(231)
在一个样本的客户机/服务器 NLM 中实现 NCP 扩展器	(231)
服务器一方	(231)
客户机一方	(253)
小结	(284)
第八章 存储管理和保护	(285)
Netware 4.0 存储管理	(285)
存储管理和处理程序	(285)
存储管理服务控制命令	(287)
NetWare 4.0 存储保护	(288)
环保护	(288)
范围保护	(288)
开发者选项	(290)
存储保护演示 NLM: Prodemo	(290)
存储保护命令	(290)
调试(debugger)命令	(292)
小结	(294)
第九章 编写与多种 NetWare 版本兼容的 NLM	(295)
向下兼容性:使用目录库和装订库	(295)
Broker NLM 如何动态地输入目录库服务 APIs	(296)
Broker NLM 是如何工作的	(301)
本章讨论的 C 接口程序	(302)
建立一个基于装订库的客户	(307)
建立一个基于目录库的客户	(324)
删除一个客户	(331)
在目录库中生成一个 Broker 目标(Broker Object)	(332)
向下兼容性:合适的 APIs 和预处理器指令	(346)
在 ENGINE. H 中的预处理器语句	(346)
在 ENGINE. C 中的预处理器语句	(347)
编写一个运行于 NetWare SFT III 上的 NLM	(351)
镜像服务器的工具:IOEngine 和 MSEngine	(351)
IOEngine 支持 C 语言接口程序	(352)

MSEngine 支持的 NetWare C 接口程序	(353)
小结	(353)
第十章 NETWARE 可移植性	(354)
字节顺序和数据校准	(354)
指导 1: 遵循标准的技术说明编写程序(ANSI C)	(355)
指导 2: 避免对数据目标值的描述作出假设	(355)
在结构中引用元素	(356)
将“字节”指针拼接成“字”或“长字”指针,将“字”指针拼接成“长字”指针 ..	(358)
把非校准数据用于库程序	(359)
对于所有的数据长度使用长度操作符	(359)
寻址数组元素	(359)
指导 3: 避免对字节顺序做假设	(359)
指导 4: 使用宏指令消除机器字节顺序特性	(361)
指导 5: 避免将若干字节组合成一个长字或字	(384)
指导 6: 避免通过长字指针分配/初始化字节	(386)
指导 7: 避免对堆栈的用法和堆栈的增加方向做假设	(387)
指导 8: 不要假设缺省使用的编程语言或计算机	(388)
使用 INT	(388)
定义范围与使用局部和全局变量及函数	(389)
使用变参方式来实现那些需要参数数量是变化的函数	(389)
位图	(389)
指导 9: 对依靠环境的程序进行标识、分离和参数化	(390)
指导 10: 在每个程序中包括头文件 Portable. H 文件	(391)
其它方面的指导	(391)
尽可能地将特殊的用户接口从程序中分离出来	(391)
用许多计算机上的各种不同的编译器来测试程序	(391)
让有不同背景的人来检查程序	(391)
小结	(392)
第十一章 名字空间、数据流、扩展属性	(393)
什么是名字空间?	(393)
什么是数据流?	(394)
什么是扩展属性?	(395)
NetWare 的名字空间机制的概要	(395)
文件工具(File Engine)接口的综述	(396)
AFP(Apple Talk Filing Protocol)API 综述	(398)
小结	(399)

第一章 NLM 基础

NetWare 可装入模块(NLMs)是 NetWare 操作系统各种版本的基本部分。NetWare 的许多基本服务是以 NLMs 的形式提供的。由于 NLMs 在 NetWare 环境中的重要性,对每一个 NetWare 用户来说,了解 NLMs 能干什么,是非常有意义的。对关心网络程序设计的程序员来说,了解 NLMs 就不仅是有意义,而且是必不可少的。

本章对于想掌握 NLMs 程序设计的程序员是一个重要的起点,另外还介绍了本章之后各章的很多内容。NetWare 和 NLM 程序设计的特点和能力的一般解释都将在本章给出,当然更详细的描述将在以后各章出现。在你阅读本章的时候,就好像愉快地和 NetWare 在一起旅行。NetWare 能做的事情将特别介绍给你,NLM 的意义和说明也将给出,从而使你能认识和辨别需要 NLM 做的事情。

还要介绍 NLM C 语言应用程序接口(API),你可以识别和定位多种文件和函数。并非本书的所有接口在本章都提到,但在你阅读完本章后,你将知道选择接口的重要性。程序设计工具(编辑、编译、链接和调试程序)也将介绍给你,并且你可以得到如何使用它们的一般说明。

▶ ◀

熟悉 NetWare 操作系统环境

NetWare 操作系统(NOS)是一个特殊的操作系统。在微机上运行的许多操作系统是通常意义上的操作系统,NetWare 不是这种类型的操作系统,这个区别对 NLM 程序员来说是重要的,因为 NLM 程序非常接近 NetWare。

普通操作系统意味着要提供这样的服务,即满足许多不同类型应用的需要。很多时候操作系统将强加限制并且迫使依从一组普通规则。有时认为这是合乎需要的,因为它是使各种应用存在于这个普通环境中的折衷办法。应用程序的编写可以和其它的协作程序没有什么关系,也不用考虑计算机资源,如内存和 CPU 的共享。

在普通操作系统中,应用程序和协作程序的联系是由操作系统来完成的。应用程序有时试图以它自己产生的方法来做某些事情,而不愿由操作系统来做这些事情。这就产生了一个保护层以使操作系统和应用程序间不易于引起问题。

NetWare 是一个专用操作系统,它从一开始到非常完善时都是为网络服务设计的。特别是对客户/服务器应用非常有效。对于表格和字处理等应用则最好是在普通操作系统上运行。而对于打印服务、数据库服务、通信服务或任何提供资源共享的服务都最好在 NetWare 上运行。

NetWare 操作系统中的每行代码都有很高的效率。为最有效地进行操作,可以假定 NLM 应用是在一个专门为其设计的环境中运行。

NetWare 增加效率的一种方法是以非优先权的方式操作。这就是说应用程序和操作系统运行的大部分时间里不可能由其它应用程序中断。由于应用程序在使用系统资源时都希望非常友好,因此,有时把它叫作“好人环境”。事实上,NLM 并非定时地将 CPU 的控制权交给其它的 NLMs 运行,而只有一个 NLM 在运行。因此在这个非优先级环境中理解独占 CPU 的含义是非常重要的。

由于 NetWare 操作系统是非优先权的,因此你不必太操心监控运行中的 NLM 的情况。只需记录正在运行的 NLM 的总开销。NLM 以什么优先级运行,运行多长时间等对 NetWare 是没有多大关系的。NetWare 记录这些东西,除了在极端情况下,它并不进行强加限制。在大多数情况下,NetWare 操作系统相信所有的 NLMs 是“好人”,这些 NLMs 通过频繁的暂停控制权来共享 CPU,这使得 NetWare 更有效地运行。

NetWare 本身同希望运行的进程队列有关,并且提供了从一个进程到下一个进程允许平滑转换的机制,NetWare 操作系统同以下内容有关:

- ▶ 内存管理
- ▶ 调度
- ▶ 通信协议
- ▶ NetWare 核心协议(NCP)进程
- ▶ 路由功能
- ▶ 高速缓存系统管理
- ▶ 文件系统访问
- ▶ 文件和记录锁
- ▶ 事务处理跟踪
- ▶ 设备驱动程序接口

利用 386、486 的编址和多任务能力,NetWare 也可以在 CPU 保护模式下运行。在保护模式中内存可以作为一连续的地址范围编址。这种连续的内存模块使内存的分配和管理更有效和更灵活。不需要转换内存段地址。因为所有内存都在一个段中,内存区域的分配仅由可用内存的大小来决定。在实模式(与保护模式相反)下运行时,由于 64K 段大小的限制,内存分配被限制为 64K。

保护模式的另一个优点是它具有并行运行多道程序的能力。这就是常说的“多任务”。在 NetWare 环境中,每个任务或进程被叫做“线程”,NetWare 通过允许和鼓励多线程程序设计来充分利用多任务,在 NetWare 中找出一个单线程 NLM 是非常罕见的。

了解 NLM 是什么?

NetWare 可装入模块(NLM)是一个程序,它可以和 NetWare 服务器或路由器在同一台机器上运行。了解 NLM 是什么东西的简单方法,就是看看 NLM 是怎样工作的。在继续对

NLM 的考查之后,你将对 NLM 的来龙去脉及相关的东西有一定的了解。

NLM 能做什么?

在可执行的结构中,NLM 是在 NetWare 服务器上通过 NetWare 操作系统可装入、可执行的文件。除了它必须有专门的扩展名 *.NLM、*.DSK、*.NAM、*.LAN 之外,可以像其它 DOS 可执行文件一样命名。具有 NLM 扩展名的 NLMs 通常提供一种服务或实用程序功能。这种类型你多半都可进行编程,具有 *.DSK 扩展名的 NLM 是磁盘驱动器程序,*.NLM 文件是名字空间 NLMs,*.LAN 文件是网络接口卡驱动程序。

NLM 的源代码看起来类似于其它的 C 语言源代码。NLM 需要的文件包括:

- ▶ PRELUDE.OBJ
- ▶ 定义文件(*.DEF)
- ▶ C 语言源文件

PRELUDE.OBJ 文件和 NLM 链接在一起以提供起动码和 ANSI C 标准函数。处理运行时间(run-time)参数和识别 main()过程。

NLM 定义文件是正文文件,它包括用于指示编译和链接指向的内容以及一些有关 NLM 的综述信息。

怎样编译一个 NLM?

NLM 的编译非常像其它的程序。为使一个普通程序进入运行的 NLM,你必须链接前述的 PRELUDE.OBJ 并在定义文件中指定所需的参数,Watcom C 编译和链接程序通常用于编译 NLMs,也可用于编译本书的范例代码。还有一个特殊的链接程序叫做 NLMLINK,将在后面讨论。另外还有几个非常好的来自 SDK 的 NLM 代码例子,你可在闲暇之余试一试。事实上,如果你还没有这样做的话,可以要求 SDK 安装编译程序和范例代码,现在你可以编译和运行范例代码,然后回过头来继续本书的内容。

NLM 程序设计所需内容

在通常情况下关于 NLM 你需要了解的一件事是,它靠在 NetWare 操作系统边上运行。即他们访问相同的大部分内存,使用同一 CPU。你可以分配自己的堆栈、变量、内存,也可以有一组好的函数库以提供你可以执行的许多功能。在 NLM 里你可以从 NetWare 操作系统中请求服务和获得信息。在下一章可以很好地观察一下这些内容。

▶ ◀

熟悉 NetWare NLM C 接口库函数