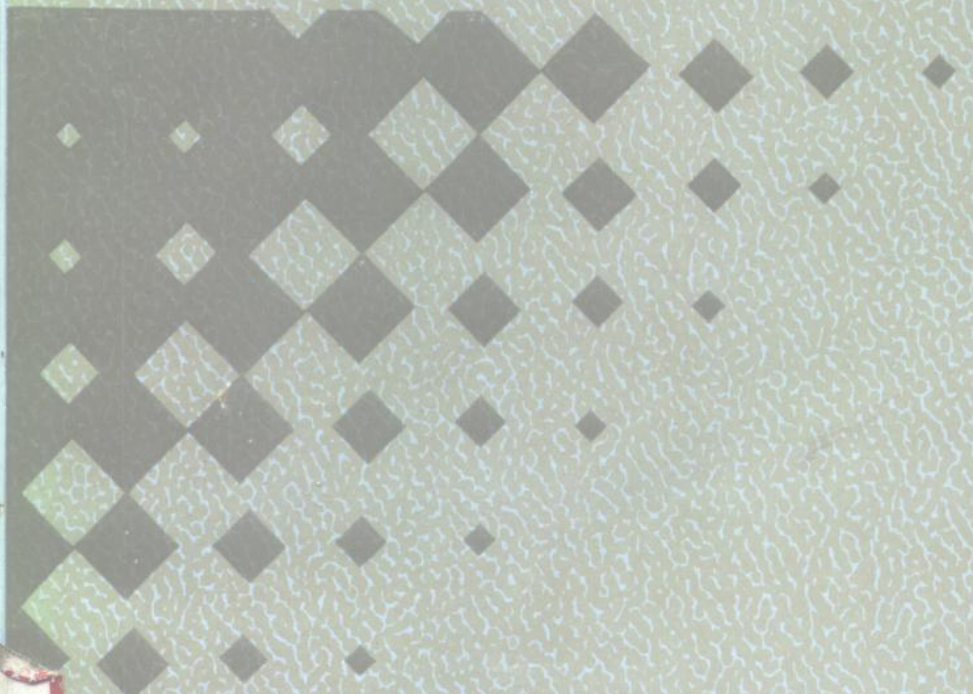
计算机实用网络编程

汤毅坚编著　　刘福滋审



人民邮电出版社

计算机实用网络编程

汤毅坚　编著
刘福滋　审

人民邮电出版社

登记证号(京)143号

内 容 提 要

本书是一本实用的计算机网络软件开发参考书。全书选编了 NetBIOS、Novell Netware 和 UNIX 三种环境下的网络编程，分别介绍了它们的基本概念、编程技巧、基本程序和应用程序等，并给出了一些实用的网络服务程序。

本书语言简洁明了，尽量回避抽象的概念叙述和复杂的理论推导，代之以大量的图表和实例，凡具有一定软件基础的读者，阅读和使用本书都不会感到困难。

本书可供从事计算机网络和数据通信工作的工程师以及高等院校相关专业的师生参考。

计算机实用网络编程

汤毅坚 编著

刘福滋 审

责任编辑 王若珏

*

人民邮电出版社出版发行

北京东长安街 27 号

中国铁道出版社印刷厂印刷

新华书店总店科技发行所经销

*

开本：787×1092 1/16　1993 年 3 月　第 一 版

印张：35　页数：280　1993 年 3 月　北京第 1 次印刷

字数：885 千字　　印数：1— 6 000 册

ISBN7－115－04811－8/TN・571

定价：22.00 元

编者的话

计算机网络在我国已经蓬勃发展起来。然而，当人们真正要在网络上做一些工作，编写一些程序的时候就会发现，这方面的参考资料，特别是实用的资料太少了。这个问题在美国也同样存在。根据我们的工作实践，参考了收集到的一些近年来出版的网络参考书和有关文献编写了本书，目的是为我国从事计算机网络开发的工程师们提供一本实用的参考书。

考虑到我国目前的实际情况和美国在网络方面发展的大趋势，本书选编了 NetBIOS、Novell Netware 和 UNIX 三种环境中的网络编程。NetBIOS 是 80 年代初为 IBM PC 局域网设计的通信程序接口，在 IBM 推出令牌环局域网时又进行了扩充。由于它的编程比较简单，直到现在仍拥有大量的用户。Netware 是 Novell 公司近年来推出的局域网操作系统。它可以在很多种硬件上运行，为用户提供丰富的服务支持，而且有一整套全面的保密措施，从而成为目前美国市场上最受欢迎的网络软件。UNIX 是目前在计算机和工作站领域中占主流地位的操作系统。在这个环境中进行网络编程可以充分利用 UNIX 的优越性能，提供高质量的网络服务。

本书共分三篇。第一篇介绍在 NetBIOS 支持下的编程，共分 5 章。主要介绍 NetBIOS 的基本概念和功能、编程技巧和一些基本程序，并给出了两种基本通信形式(数据报和会话)的通信程序实例。

第二篇介绍 Novell Netware 环境下的网络编程，共分 4 章，在介绍 Netware 的基本概念和两种基本通信协议 IPX 和 SPX 之后，介绍了 IPX 和 SPX 的编程技巧、基本程序和应用程序，以及 IPX/SPX 的功能调用和 6 种扩充 DOS 服务。

第三篇是本书的重点，详细介绍了 UNIX 环境中的网络编程，共分 9 章，分别介绍 UNIX 和网络通信的基本概念、两种进程间通信手段、UNIX 的网络库程序和实用的网络服务程序等。

本书在语言上力求简捷明了，尽量回避抽象的概念和复杂的理论，代之以大量的图表和实例。因此，凡具有一定软件基础(主要是 C 语言、DOS 和 UNIX 操作系统)的读者，阅读和使用本书都不会感到困难。由于编者水平所限，不足之处在所难免，恳请广大读者多加指正。

哈尔滨科学技术大学
汤毅坚
1992.4

目　录

第一篇　NetBIOS 支持下的编程

第一章　概　　述

第一节　引　言

NetBIOS (Network Basic Input/Output System，网络基本输入/输出系统)是一种用于数据交换的专用程序，是计算机设备和应用程序进行通信的软接口。1984 年 8 月，Sytek 公司推出了为 IBM 个人计算机设计的第一个网络—IBM PC Network。这是一个以同轴电缆作为通信媒介的计算机局域网(LAN)，数据传输速率为 2Mbit/s，采用当时很流行的以太网通信协议，即带碰撞检测的载波侦听多路访问(CSMA/CD)协议，也就是后来颁布的 IEEE 802.3 标准。在 IBM PCNet 局域网适配器(LANA)接口板上安装了 2 片 8k 字节的 ROM，含有 LANA 的初始化程序、协处理器程序、存储器接口程序和诊断程序，还含有对网络通信进行管理的程序。由于这部分程序可以看作是 IBM 基本输入/输出系统(BIOS)的扩展，因此，人们通常把它叫做 NetBIOS。

1985 年 10 月当 IBM 公司推出它的另一个局域网—IBM 令牌环(Token Ring)的时候，推出了相应的网络管理程序，称为 NetBIOS 扩充用户接口，简称 NETBEUI.COM。这个程序模块和另一个模块 TOKREUI.COM 一起，为令牌环提供数据链路层的控制功能(DLC)。NETBEUI.COM 把 NetBIOS 的命令翻译成 DLC 的命令，供接口执行。

现在，IBM 网络支持程序提供“真正的”NetBIOS。这种程序可以通过各种 PC-DOS 设备驱动器，来支持所有的 IBM 局域网适配器。从 1987 年以后，工业界颁布了 NetBIOS 的扩充版本 TCP/IP.NetBIOS，更加扩大了 NetBIOS 的应用领域。现在，NetBIOS 可以支持各种著名的通信协议，如 TCP/IP、MAP/TOP、XNS、IEEE 和 OSI 等；而且不仅适合于 DOS 环境，还可以用在 UNIX 和 OS/Ⅱ环境。本书对于 NetBIOS 的讨论仅仅局限于 PC-DOS。

NetBIOS 作为一种网络支撑软件，在 ISO 的开放系统互连(OSI)模型中处于什么位置呢？对此有几种不同的说法。比较流行的说法是，NetBIOS 高于会话层，而又低于表示层，界于两者之间，如图 1.1 所示。把 NetBIOS 安排在较高的层位，有利于把用户的应用程序和 NetBIOS 之间的通信同具体的网络实现方式隔离开来。比如，两个应用程序可以通过 IBM PCNet 适配器通信，而下层的通信则通过会话管理协议(SMP)来实现；这两个程序也可以通过 IBM 令牌环

适配器通信，而下层通信则采用数据链路控制(DLC)协议。上层的通信和下层的具体实现完全独立。这样，通用的 NetBIOS 应用程序就可以在广泛的通信环境中兼容，或者很方便地移植。

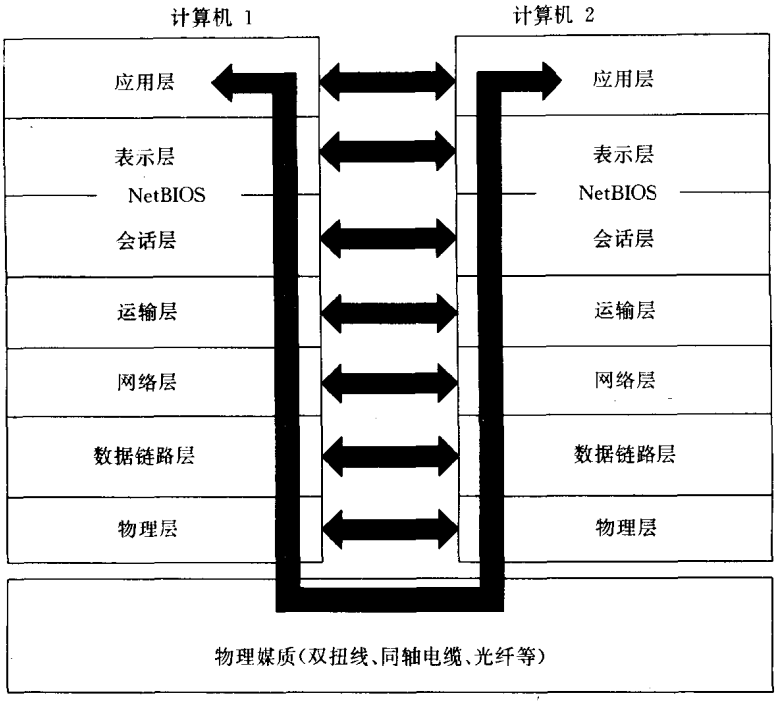


图 1.1 NetBIOS 在 OSI 模型中的位置

第二节 NetBIOS 的基本概念

1. 命令

NetBIOS 的功能是由执行一系列命令来完成的。大多数 NetBIOS 命令都有 wait 和 no-wait 两种形式。在采用 wait 形式时，NetBIOS 要等到该命令完成之后才返回你的程序。如果采用 nowait 形式，那么又可以有两种选择：一种是采用轮询(poll)方式，即反复询问，直到该命令完成；另一种是把你的一个程序的地址告诉给 NetBIOS，让它在完成命令时唤醒该程序。在采用后一种非等待方式时，NetBIOS 立即返回你的程序，并且带回“立即返回代码”；而当命令执行完时，又返回一个“最终返回代码”。你的程序可以根据这两个代码确定该命令是否成功执行。

2. 名字

NetBIOS 是一种按名字工作的系统，每一个网络工作站和服务器都有一个(或几个)名字。由 NetBIOS 保留一张名字表，每个名字都为 16 字节长。每个节点还有一个永久节点名，它是在网络适配卡上的 6 字节网络地址前面再添加 10 个字节 0 构成的。网络中各个节点的永久节点地址是互不相同的。

在名字表中除了每个节点的名字外，还可以有小组名。几个工作站可以采用同一个小组名，表示它们处于同一组内。在进行通信时，信息的源地址和目的地址都是用名字来表示的。它们可以是永久节点名、用户增加名、或者小组名。

在为一个节点指定名字时注意不要用 * 号或二进制 0 开头，也不要用 IBM3 个字母开头，因为这些名字都是保留名。为了保证名字表中的名字都是互不相同的，在往名字表中增加新名字时 NetBIOS 首先检查本节点的名字表，如无重复，则向其他各节点发一个广播分组，各节点均应检查名字表中是否有此名字。只有不重复的名字才是合法的名字。

3. 数据报和会话

数据报是一种无连接的服务，即各个数据报之间互相独立，单独传送。数据报的目的地址可以是节点名、小组名，也可以是网络中的所有节点。每个数据报中信息记录长度最多为 512 字节。

在网络上任何两个名字之间可以建立一个会话，也可以建立多个会话，还可以在同一个工作站上的 2 个名字之间建立会话。在会话进行过程中，发出的所有报文都能准确地投递到目的地。报文记录长度上限为 65 535 字节。

	长度(字节)
命令 ID	1
立即返回代码	1
本地会话号	1
网络名字号	1
数据缓存地址	4
数据缓存长度	2
对方名字	16
自身名字	16
接收超时	1
发送超时	1
POST程序地址	4
适配器号	1
最终返回代码	1
保留字段	14

图 1.2　网络控制块格式

4. 网络控制块(NCB)

应用程序要调用 NetBIOS 命令，必须先构造一块网络控制块(NCB)，然后执行一次 5CH 中断。NCB 的格式如图 1.2 所示。下面我们分别介绍其中的各个字段：

1)命令 ID(1 字节)：指定所要执行的命令。其中，最高位为“1”表示非等待方式，为“0”表示等待方式；

2)立即返回代码(1 字节)：含有出错信息，在开始执行该命令时由 NetBIOS 设置；

3)本地会话号(1 字节)：在建立会话过程中，由 NetBIOS 为每一个会话设置一个编号。在要执行 Send 或 Receive 命令时要把对应的会话号放入本字段；

4)网络名字号(1 字节)：在往网络中增加一个名字(节点名或小组名)的时候，NetBIOS 会在这个字段中返回给你一个名字号。以后，在采用数据报方式(注意：不是会话方式)通信时，就可以使用这个一字节的名字号，而不必使用 16 字节的名字；

5)数据缓存地址(4 字节)：利用这个字段，可以使一个远程指针(Segment:offset)指向和发送/接收有关的数据缓存；

6)数据缓存长度(2 字节)：指定数据缓存的字节数；

7)对方名字(16 字节)：指定要通信的对方名字；

8)自身名字(16 字节)：指定你的应用程序所要使用的名字(可以是本地名字表中所列的

名字，或是永久节点名）；

9）接收超时（1 字节）：在发出 Call 或 Listen 命令时，可以在此字段中设置一个数值（以 0.5s为单位），指定 NetBIOS 最多可以等待多长时间。如不指定此值（即为 0），则表示可以一直等下去，直至 Receive 命令完成；

10）发送超时（1 字节）：在发出 Call 或 Listen 命令时，可以在此字段中设置一个数值（以 0.5s为单位），指定 NetBIOS 最多可以等待多长时间。如不指定此值（即为“0”），则表示可以一直等下去，直至 Send 命令完成；

11）POST 程序地址（4 字节）：可在这个字段中放入一个远程指针，指定在该命令完成时 NetBIOS 要唤醒的那个程序的地址。如前所述，这个字段只有在非等待方式时才有意义。如果采用非等待方式中的轮询选择项，即循环检查命令是否完成，且成功执行，那么就要把此字段设置为空（即“0”）；

12）适配器号（1 字节）：每个工作站最多可以有两个令牌环适配器。可以用这个字段指定选用哪一个：“0”表示基本适配器，“1”表示备用适配器；

13）最终返回代码（1 字节）：在命令执行过程中其值为 0xFF，命令执行完毕后设置表示执行成败的代码；

14）保留字段（14 字节）：用户不能使用。

5. 信息帧

在网络中传递的信息是以帧的形式组织的。NetBIOS 根据用户的命令、NCB 以及它所知道的有关会话和名字的情况自动生成和管理信息帧，用户不必直接过问。信息帧的种类很多，下面简单介绍几种比较重要的信息帧：

1）name query：在发出 Call 命令时，此信息帧向全网络广播，寻找网络中是否有此工作站；

2）add name query 和 add group name query：在发出 Add Name 和 Add Group Name 命令时，用来询问网络中是否已有同样名字在用；

3）name recagnized：是对 name query 的响应，表示此名字存在且准备接收；

4）add name response：是对 add name query 的响应，表示此名字已经在用；

5）session initialize：用来建立会话；

6）session confirm：表示会话已建立；

7）data：当发出 Send 命令时，NetBIOS 发出数据帧；

8）ACK：一个工作站成功地接收到 data 帧后发出 ACK 帧确认；

9）session end：当发出 Hang Up 命令时 NetBIOS 发出此信息帧，结束会话；

10）datagram：当采用数据报方式通信时用来发送数据。它和 data 帧相似，但不要求先建立会话。

6. 服务器报文块（SMB）

IBM PC 局域网程序能够把和网络有关的 DOS 功能调用筛选出来，并且经网络送往文件服务器，这就要采用一种称作服务器报文块的协议。在工作站上运行的 PC 局域网程序和在文件服务器上运行的 PC 局域网程序之间建立一个会话，并且用 NetBIOS 的 Send 命令向服务器发送一个服务器报文块（SMB），请求服务器为该工作站进行文件操作。SMB 有下列 4 种类型：

1）会话控制类：在工作站和文件服务器之间建立起会话之后，工作站向服务器发出称作

Verify Dialect 的报文块，指明在工作站上运行的 PC 局域网程序的版本。服务器接收此信息，并且向工作站提供关于服务器的信息。接着，要发送一个或几个 Start Connection 报文块，在工作站和文件服务器的各个网络资源之间建立起逻辑连接。通信结束时由工作站发出 End Connection 报文块给服务器，终止这些逻辑连接。

2)文件访问类：工作站采用这一类报文块可以远程访问服务器硬盘中的各种文件，就象访问本地的硬盘一样。主要功能有：

a. 建立目录/撤消目录；

b. 建立、打开和关闭文件；

c. 读/写文件；

d. 为文件更名/删除文件；

e. 搜索文件；

f. 获取/设置文件属性；

g. 封锁记录。

工作站把这些远程操作命令转变为报文块送给服务器去执行，本地的 PC DOS 根本看不到这些命令。

3)打印服务类：采用这一类报文块可以把文件送到打印服务器排队，还可以获取打印服务队列的状态信息。工作站可以建立一个假脱机文件，写入数据，并关闭该文件；也可以要求服务器返回 Print Queue Status 报文块，报告打印机队列的状态。

4)传递消息类：用来完成一些简单信息的传递，如：

a. 发送单块报文；

b. 发送广播报文；

c. 发送多块报文的开头、结尾和正文；

d. 转送用户名；

e. 获取机器名，等等。

图 1.3 给出服务器报文块的格式，下面简单介绍其中的各个字段：

1)报文类型(1 字节)：指明本报文块中所含报文的类型；

2)服务器类型(3 字节)：指明本报文块接收端的服务器类型；

3)功能 ID(1 字节)：表示本报文块所代表的网络请求的类型；

4)返回差错类型(1 字节)：在发生差错时指明发生差错的位置；

5)关键差错类型(1 字节)：给出在服务器中发生的关键差错的信息；

6)返回代码(2 字节)：指明命令完成的状态；

7)方向代码(1 字节)：指明本报文块是对服务器的请求还是给工作站的响应；

8)保留字段(14 字节)：用户不能使用；

9)网络路径 ID(2 字节)：含有计算机名、路径名和文件名，指明工作站和特定的服务器资源之间的逻辑路径；

10)进程 ID(2 字节)：用来标明产生本报文块的 DOS 功能调用所在的程序；

11)保留字段(4 字节)：用户不能使用；

12)参数计数(1 字节)：说明下一字段中有几个参数；

13)参数字段：它的意义和长度随着报文块的功能而异；

14)缓存长度(2 字节)：指明本报文块中缓存字段的字节数；

	长度(字节)
报文类型	1
服务器类型	3
功能ID	1
返回差错类型	1
关键差错类型	1
返回代码	2
方向代码	1
保留字段	14
网络路径ID	2
进程ID	2
保留字段	4
参数计数	1
参数字段	不定
缓存长度	2
缓存字段	不定

图 1.3　服务器报文块(SMB)的格式

15)缓存字段:用来放置和各功能相关的数据,其格式可变。除一般数据外,本字段还可包含一个或多个可变长度的结构。每一个结构第一个字节指明该结构的类型,接下去 2 个字节指明该结构的长度。

第二章 编程技巧

上一章提到了工作站和文件服务器之间的信息交换。由一个工作站把信息写入文件服务器，然后由其他工作站去读取，这只是网络中，工作站之间信息交换的一种方式，但并不是最好、最方便的方式。因为使用文件服务器存在一些缺点，比如速度比较慢、需占用磁盘空间、要经常对文件系统进行清理维护等，而且要求用户具有一定的读写访问权才能访问文件系统。再有，如果 A 站要给 B 站发信，它把信息作为一个文件写入文件服务器，那么 B 站如何得知 A 站已给它发信呢？因此，在很多场合采用 PC 和 PC 之间的直接通信往往更为简捷明了。PC 和 PC 之间通信时，最常用的通信协议是 NetBIOS、IPX 和 SPX。

第一节 设计对话

在局域网中进行的通信，有的是很简单的实时交谈、电子邮件，有的是很复杂的分布式处理系统的任务分配，但不论是哪一种通信，都可以把它看作是两个或多个有关进程之间的对话。我们把这种对话想象成一个特殊类型的文件：接收一个报文相当于从这个文件中读出一个记录；而发送一个报文则相当于往这个文件中写入一个记录。对于报文的格式和内容的定义就像定义一个文件中的几个记录一样。

1. 简单对话

在计算机通信中最简单的形式是实时的交谈——每个记录中只包含一次击键。当交谈的一方按下一个键时，交谈程序一方面把相应字符显示在击键方工作站的屏幕上，另一方面则把含有这个字符的记录送往对方；同样，当对方回答时，每按下一个键则送来一个记录，并把该记录所含的字符显示在这一方的屏幕上。在设计交谈程序时最麻烦的问题是要管理好屏幕上的两个窗口：上半部是接收窗口，显示对方发送来的字符；下半部是发送窗口，显示本站向外发送的字符。

电子邮件的种类很多，复杂程度差异也很大，但它们所使用的记录格式是大同小异的。通常都有一个报头，其内容包括发信人姓名、收信人姓名、地址以及本报文的主题等，其作用相当于普通邮件的信封。记录中除了报头之外，通常就只有正文了，就像是信封中所装的信一样。

2. 复杂对话

局域网如果用来连接分布式数据处理系统，那么情况就复杂多了。系统中各个工作站可能按处理步骤来分工，比如 A 工作站进行第 1 步处理，B 工作站进行第 2 步处理，C 工作站进行第 3 步处理，……，那么，A 工作站首先接收 1 个记录，进行第 1 步处理，并且把处理结果作为

1 个记录送给 B 站进行第 2 步处理，然后又把结果送给 C 站，……，这时，系统是以串行的模式工作。

系统也可以以并行模式工作，即每个工作站都具有相同的功能，运行相同的程序，就像在银行、邮局中多个服务员为 1 个顾客队列服务的情况一样，当一个服务员服务完当前一个顾客后又从顾客队列中取出排头进行服务。在这种服务模式中，记录的传递主要是上一步与下一步之间的传递。在同一步上几个工作站之间则基本上不传递信息。

对于真实的分布式数据处理系统，情况可能要复杂得多。各个工作站的功能往往是动态指派，而不是一成不变的；而记录的交换也不只是工作站之间，也可能是同一工作站的不同磁盘存储区之间的交换。在这种情况下就要很仔细地设计对话。

第二节　设计要点

在设计两个或多个工作站之间对话的时候，要注意处理好以下几个问题：

1. 选择通信方式

在上一章谈到，网络中的通信可以采用数据报方式，也可以采用会话方式。

会话方式的特点是由通信协议保证每一个报文能够正确地送到目的地。如果报文在传送过程中发生了差错，则通信协议会自动地进行重发，加以更正。而且，如果你发出的报文不止一个，那么通信协议还能够保证接收到的报文次序和发送时的次序一致。这些保证都是对用户应用程序透明地进行的，用户完全不需操心。但是，会话方式的主要缺点是过程比较复杂，在发送报文之前先要建立会话，送完报文之后又要撤除会话。因此，会话方式比较适合于点到点的多次往复式的对话。

数据报的特点正好相反。它是把每一个报文作为一个独立的数据报处理，从而免除了建立和撤除会话的麻烦。但正因为如此，它不能保证接收到的报文次序和发送次序一致，甚至不能保证每一个报文都能正确无误地到达接收方。在报文丢失或发生差错时，系统不向发送方提供出错信息。如果你所要设计的对话环境由一系列互不相关的报文组成，或者每次对话都很简单，那么采用数据报可以大大加快对话的速度。另外，如果用数据报传送比较重要的报文，为了防止差错和丢失，可以在应用程序中为报文加上查错和序号功能。

NetBIOS 既支持会话方式，又支持数据报方式。NovellNetware 中的 IPX 协议只支持数据报，SPX 协议（从 Netware 2.1 版开始出现）可以支持会话方式。

2. 选择命令返回方法

在第 1 章曾谈到，执行 NetBIOS 命令可以采用等待方式和非等待方式。对于非等待方式又可以有两种不同做法：一种是轮询方法，即循环检查网络控制块（NBC）中的最终返回代码字段，当它的值从 OXFF 变为其他值时表示该命令执行完毕；另一种是异步事件处理方法，即告诉 NetBIOS，在当前命令执行完毕后转向存放在 NBC 的 POST 程序地址字段中的程序指针，继续执行一个指定的程序段。需要指出的是，这两种做法虽然难易程度不同，但都是在后台进行。对于前台操作，都是立即返回用户程序，不必等待命令完成。它们的主要差别在于对网络

负荷的影响。比如，一个工作站发出一条命令，要求另一个工作站执行一项比较费时的任务。如果这个工作站采用轮询方式，持续不断地询问对方是否完成，就会极大地增加网络的负荷。这时，尽管轮询比较容易编程，那也不是一种好的选择。

3. 分组长度的限制

不论何种通信协议，对于在网络上收发的分组长度都有一个上限。比如，对于 NetBIOS 的数据报服务，上限值通常为 512 字节；对于 NetBIOS 的会话服务，采用 Send 命令时上限值为 65535 字节，而采用 Chain Send 命令时可以同时传送两个缓存的内容，字节长度的上限也加倍。

处理分组长度上限通常有两种做法。一种是按照长度限制 把数据截为若干段，对每一段进行一次发送，到了接收端再把它们装配起来复原。这样做虽然需要增加一定的编程工作量，但好处是可以把程序主体和分组上限问题分开来处理。

另一种做法是专门按照长度上限的规定来设计数据记录的格式，这样每一个报文都可以毫无困难地发送。但是，这样做的缺点是一旦在已有的记录中增加数据字段的话，只能重建记录。

4. 差错和超时

在以非等待方式执行命令时，NetBIOS 首先返回一个立即返回代码，并且在命令执行完毕后再返回一个最终返回代码(对于等待方式则只有最终返回代码)。附录 A.1 列出了返回代码，可以看出，当命令执行成功时，返回代码为 OOH，返回其他值均为差错。为了使程序顺利运行，差错处理是必不可少的。当命令执行完毕后，首先要检查返回代码，如果返回值为“0”，说明命令执行成功，可以继续执行下一条命令；如果返回值不是“0”(也不是 FFH)，说明出现差错，就要根据差错代码确定差错性质，并且作出相应处理。除了一些致命的差错之外，通常进行差错处理之后还可以继续执行后面的命令。

当一个工作站向另一个工作站发出分组之后，经过多少时间才能收到对方的应答呢?这个问题在很大程度上取决于网络的类型和覆盖面积，也取决于网络上通信的繁忙程度。对于像以太网、令牌环这样的局域网来说，由于地域上分布比较集中，网络的数据传输速率又比较高，因此一个 512 字节的报文通常只需 1～2ms 就可以到达目的地。我们可以参考这个数量级来设置超时时限。如果向某一个工作站发信，连续发生超时出错，那么很可能不是因为网络过于繁忙，而是接收站发生了故障，或者已经关机。

第三节　NetBIOS 基本程序

• 1. 调用 NetBIOS

要调用一个 NetBIOS 功能，需要做以下三件事：

1) 构造一个网络控制块(NCB)，包含所有需要告诉 NetBIOS 的信息。前面第 1 章中已介绍过 NCB 的格式(参阅图 1.2)，下面定义的这个结构和 NCB 格式完全一一对应：

```
typedef unsigned char byte;
typedef unsigned int word;
/* Network Control Block (NCB) */
typedef struct
    {
    byte NCB_COMMAND;
    byte NCB_RETCODE;
    byte NCB_LSN;
    byte NCB_NUM;
    void far *NCB_BUFFER_PTR;
    word NCB_LENGTH;
    byte NCB_CALLNAME[16];
    byte NCB_NAME[16];
    byte NCB_RTO;
    byte NCB_STO;
    void interrupt (*POST_FUNC)(void);
    byte NCB_LANA_NUM;
    byte NCB_CMD_CPLT;
    byte NCB_RESERVE[14];
    }
    NCB;
```

在编程时应该对每一种操作定义一个网络控制块，这样做比多个操作合用同一个 NCB 容易管理；

2)把网络控制块的地址写入 ES:BX 寄存器对，作为指向该控制块的远程地址指针；

3)执行 5CH 中断。执行中断时，NetBIOS 自动进入 ES:BX 寄存器指定的地址，读出网络控制块，得知一切有关该操作的信息。

由上述分析可知，网络控制块只是存放信息的存储单元，它本身并不通过网络传送。下面是设置指针和执行中断的程序段：

```
void NETBIOS (NCB far *ncb_ptr)
    {
struct SREGS sregs;
    union REGS regs;

sregs.es = FP_SEG(ncb_ptr);
regs.x.bx = FP_OFF(ncb_ptr);
int86x(0x5c, &regs, &regs, &sregs);
}
```

2. 定义 NetBIOS 命令

在程序中还应该包括下面这段程序段，对所有 NetBIOS 命令(包括 WAIT 和 NO-WAIT)定义其操作代码(均为十六进制)：

```
#define RESET                    0x32
#define CANCEL                   0x35
```