



清松电脑系列丛书

UNIX网络原理与应用

欧宗生 编著



清华大学出版社

UNIX 网络原理与应用

欧宗生 编著

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号:01-96-1424 号

UNIX 网络原理与应用

欧宗生 编著

JS218/07

本书中文繁体字版(原书名为洞悉 UNIX 网路原理篇)由台湾和硕科技文化有限公司出版,1996。本书中文简体字版经台湾和硕科技文化有限公司授权由清华大学出版社与北京清华松岗电脑信息有限公司合作出版,1997。任何单位或个人未经出版者书面允许不得用任何手段复制或抄袭本书内容。

本书针对 UNIX 系统上一些常使用的网络系统,作了完整而详细的介绍。先从网络结构开始,到应用程序界面(API)的使用,最后介绍此系统的操作方式和一种网络系统。

全书共七章。主要内容包括 UNIX 系统及其网络系统的特性与功能,计算机网络及其 TCP/IP 协议,以及 XNS、RPC、NFS、RFS 等网络系统及其管理。

本书内容翔实、由浅入深、专业性强,适合于计算机中、高级用户学习使用,同时也是广大 UNIX 网络用户的一本较好的指导书。

版权所有,翻印必究。本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

出版者: 清华大学出版社(北京 清华大学校内,邮政编码:100084)

责任编辑: 童隆裴

印刷者: 北京市王史山胶印厂印刷

发行者: 新华书店总店北京科技发行所

开本: 787×1092 1/16 印张: 10.5 字数: 240 千字

版次: 1997 年 7 月第 1 版 1997 年 7 月第 1 次印刷

书号: ISBN 7-302-02592-4/TP·1324

印数: 0001—5000

定价: 22.00 元

序 曲

从开始接触 UNIX 系统迄今已超过 7 个年头。在整个学习过程中,有几本优秀的英文读物对笔者帮助最大,例如 Marc J. Rochkind 的《Advanced UNIX Programming》、Maurice J. Bach 的《The Design of The UNIX Operating System》。这两本著作都非常好,可谓关于 UNIX 系统的经典之作,值得有心钻研 UNIX 的人士一读再读。但是对国内的部分读者来说,语言本身可能会形成一种障碍,使得学习效果大打折扣,但若看翻译的作品,不是被那种生吞活剥式的“英译中”给弄得大叹不如看原文,就是被同类书中五花八门的译名给整得晕头转向,于是乎学 UNIX 难,中国学生学 UNIX 更难。

而 UNIX 本身的内容极其丰富,所涵盖的范围从网络、数据通信、操作系统理论、电脑绘图到电脑系统安全维护等等,几乎没有一个人能对这么广阔的领域样样精通,也不可能有一本书能完整地介绍这么庞大的范畴。当然 AT&T 那一套十七本的使用手册是个不坏的参考资料,但由于它是一本英文著作,且实例少,加上它所定位的读者群是有相当经验的电脑专业人员。于是对初学者而言,它便成了“有字天书”,令人望而生畏。

和硕科技深深了解上述中国同学的学习困难,也看好 UNIX 系统未来的远景,因此结合了一群有实际经验的国内外学人依照个人的专长,推出 UNIX 相关著作,使读者能清楚地一窥 UNIX 的全貌。

这些著作推出后,读者反应之热烈令人惊叹,而其中有许多专家如美国喷气推进实验室(JPL)的王清佑先生、安源公司的黄家华先生等都给了我们许多的建议,因此我们对其作了相应的修改,希望能获得读者的喜爱。

做任何事都需要有一种理想与使命感在背后支撑,这样才能坚持信念勇往直前,也才会越挫越勇绝不退缩。和硕科技的工作人员就是仰赖这份信念,奋力不懈。希望读者在发现我们的优点时,不要吝于给我们掌声,在发现我们的缺点时,也请给我们中肯的建议。

电脑虽然是科技性的产品,但是和硕科技却将它视为文化的一种,因为唯有如此,才能使信息工业真正达到普及化、全民化。不管知识也好、技术也好,唯有让它走入大众,才能真正生根发芽、开花结果。

走在时代的后面是可悲的,走在时代的前端却又是孤独的,和硕宁可千山独行不失特色,也不愿随波逐流全无格调,只盼在蓦然回首之际能听到读者的肯定,即使是寥寥数声也足以让我们雀跃三尺,因为我们深知古来圣贤皆寂寞!

自序

在信息爆炸的时代中,信息交流变得极为频繁。人们在处理日常事务时,常用电脑来代劳,而电脑间通信的问题遂成为一个重要的课题。电脑与电脑间若不经由网络连线来交换信息,就好比独居深山中的野人与世隔绝,文明的进步将因此而减缓。

UNIX 系统给这个世界提供一个开放的空间,让不同的机器和不同的人群,通过它集成在一共同的国度中,这就是为什么“开放式系统”被世人所热爱的原因。在这个开放式的体系结构上,人们可以充分发挥自己的才华,设计出通用性强的产品。这样一来,不仅人与人之间的距离缩短了,机种差异所造成的不便也降到最低。因此当我们使用电脑时,便无需担忧因距离及差异所造成的不便及疏离感。我们可以说“这个时代是网络通信的时代”。

笔者七、八年前在交大读书的偶然机会里,首次接触到“网络”这个诱人的名词,没想到它影响了笔者日后追求的方向。大三时,笔者选修了“计算机网络概论”这门课,它引导笔者进入“网络”的领域,而交大的李素瑛教授便是笔者的网络启蒙恩师,以后不论在美国求学或工作,在交大所打下的基础皆成为笔者日后发展的基础。在此笔者感谢交大一群辛苦奉献的老师们所付出的心力,并致上笔者十二万分的谢忱。

欧宗生

目 录

第 0 章 简介	1
0.1 UNIX 系统简介	1
0.1.1 Shell	3
0.1.2 文件结构	3
0.1.3 安全性	3
0.1.4 与设备无关的输入和输出	3
0.1.5 进程(process)间的通信	4
0.1.6 数据缓存(buffering)与带外(out-of-band)数据	4
0.2 UNIX 网络系统的特性及主要服务功能	5
0.2.1 层(Layering)	6
0.2.2 OSI 模式	6
0.2.3 一种简化的模式	6
0.2.4 客户-服务器(client-server)模式	7
0.2.5 网络上的主要服务功能	7
0.3 流(STREAMS)	9
0.4 本书内容介绍	15
第 1 章 电脑网络(Computer Network)	17
1.1 网络的用途	17
1.2 网络的结构	18
1.3 网络的体系结构	21
1.3.1 通信协议的层次性	21
1.3.2 各层次通信协议的基本功能	23
1.4 OSI 通信协议模式	24
1.4.1 物理层	24
1.4.2 数据链路层	25
1.4.3 网络层	26
1.4.4 传输层	26
1.4.5 会话层	27
1.4.6 表示层	28
1.4.7 应用层	28
1.4.8 OSI 中的数据传输	28

1.5	OSI 的服务功能	29
1.5.1	连接式(Connection-Orient)与非连接式(Connectionless-Orient)传输	29
1.5.2	服务的基本功能	30
1.5.3	服务功能与通信协议的关系	32
1.6	网络的实际例子	32
1.6.1	公共网络	32
1.6.2	ARPANET	34
1.7	中继器、网桥、路由器和网关	34
第2章	TCP/IP	36
2.1	何谓 TCP/IP 中的 IP	36
2.1.1	网络层的需求	36
2.1.2	网络层的功能与组织	41
2.1.3	IP 的通信协议	46
2.2	何谓 TCP/IP 中的 TCP	51
2.2.1	传输层的需求	51
2.2.2	传输层的功能与组织	54
2.2.3	TCP 的通信协议	57
2.3	UNIX 的 TCP/IP 应用接口(API)	60
2.3.1	Berkely Sockets	60
2.3.2	TLI	69
2.4	域名系统(DNS)	73
第3章	Xerox 网络系统(XNS)	76
3.1	系统结构	76
3.2	通信规格	77
3.2.1	数据链路层	77
3.2.2	网络层——IDP	78
3.2.3	传输层——SPP 与 PEX	79
3.3	XNS 的桥(bridge)	82
3.3.1	封包通信协议类型(packet type)	83
3.3.2	数据流类型(datastream type)	83
3.3.3	客户类型(client type)	83
3.4	数据缓冲区的处理策略	83
3.4.1	数据缓冲区的大小限制	83
3.5	XNS 中的应用层(application layering)	83
第4章	远程过程调用系统	84
4.1	RPC 的透明性(transparency)	85
4.1.1	参数的传递	85

4.1.2	连接(Binding)	86
4.1.3	传输层通信协议(transport portocol)	86
4.1.4	异常处理(Exception Handling)	86
4.1.5	调用的语义	86
4.1.6	数据的表示	87
4.2	远程调用语法	89
4.3	RPC 的应用接口(API)	91
4.4	RPC 对于 XDR 的使用方式	97
4.5	RPC 的通信传输	103
4.6	SUN 的 RPC 应用接口(API)的介绍	104
第 5 章	NFS(网络文件系统)	106
5.1	何谓 NFS	106
5.1.1	软件及硬件设备的要求	107
5.1.2	通信规格	107
5.1.3	VFS 与 V 节点	110
5.1.4	NFS 的虚拟文件系统	111
5.1.5	mount 的执行	112
5.1.6	文件访问	113
5.1.7	NFS 的文件结构	116
5.2	NFS 的系统管理	119
5.2.1	系统的需求与执行	119
5.2.2	系统现况报告	130
5.3	执行效率与可靠性	130
第 6 章	RFS 系统	132
6.1	RFS 与 NFS 的相异处	132
6.2	RFS 的文件使用结构	133
6.2.1	服务器与客户机	133
6.2.2	RFS 名称服务器和它的功能	133
6.2.3	软件及硬件设备的要求	136
6.3	RFS 的系统管理	136
6.3.1	域的命名(naming)	136
6.3.2	域中的主名称服务器	137
6.3.3	RFS 的建立、启动及使用	139
6.3.4	权限的检查	142
6.3.5	监听程序服务请求	149
6.3.6	RFS 系统的故障与修复	150
6.3.7	系统状况的报告	152

第 0 章 简 介

UNIX 系统是一种开放式的操作系统,适用于不同的机种,而且其本身可支持多种通信协议(protocol),并用以完成不同机器间的数据传送。UNIX 系统具有多种特色,在此我们仅就其发展的过程及网络相关的功能加以论述。

0.1 UNIX 系统简介

UNIX 操作系统是由位于新泽西州的贝尔实验室开发出来的,迄今已有许多人参与修正及改进,而其中最有名的便是 AT&T 公司贝尔实验室版 UNIX 和加州大学伯克利分校的 BSD UNIX。图 0-1 说明它发展的过程。

图 0-1 标示的时间只是约略数字,图形左边的主要图表是发展的主要路线。前期几个 UNIX 版本都是贝尔实验室内使用的。但在七十年代初期,贝尔实验室开始向大学推出这套系统。七十年代中叶的 Interactive System One(IS/1)是一项重要的里程碑——它是 UNIX 系统的第一个商业产品,Microsoft 公司的 Xenix 系统则是下一个主要商业产品。

在学术与军事领域中,BSD(Berkeley Software Distribution)4.1 和 4.2 版则是 DEC VAX 系列电脑上的标准 UNIX 系统。今天在市场上看到的许多电脑使用的 UNIX 版本,就是从 BSD 版本延伸出来的。当初开始开发 UNIX 时,大多数的电脑还是采用批处理操作方式;程序员必须先用 IBM 打卡机将程序内容打成卡片,然后用读卡机将卡片上的程序读入电脑执行,接着以打印机打出结果。这种批处理操作方式一次只能处理一个人的程序,根本无法充分利用电脑的功能及速度。UNIX 的分时操作系统和前述的单用户批处理系统相比,有两大特点:第一,它允许多位用户同时使用同一部电脑;第二,它允许用户通过终端直接与其他用户通信。

七十年代中叶,小型电脑的优势开始超过大型电脑,小型电脑在许多应用上可以较低的价格达到与大型电脑相同的功能。以功能极强的 16/32 位处理芯片,配上容量大且便宜的内存,再加上价格低廉的硬盘,促使制造商开始在微机上安装多用户操作系统。现在这些系统的价格和性能已可与小型电脑上的系统竞争。UNIX 系统原来即准备作为发展工具,提供给软件公司一个极佳的工作环境。这种标准操作系统造就了软件工业的产生,现在的软件厂商设计出的同版本产品,可在许多不同的机器上执行。只要硬件制造商的机器能执行 UNIX

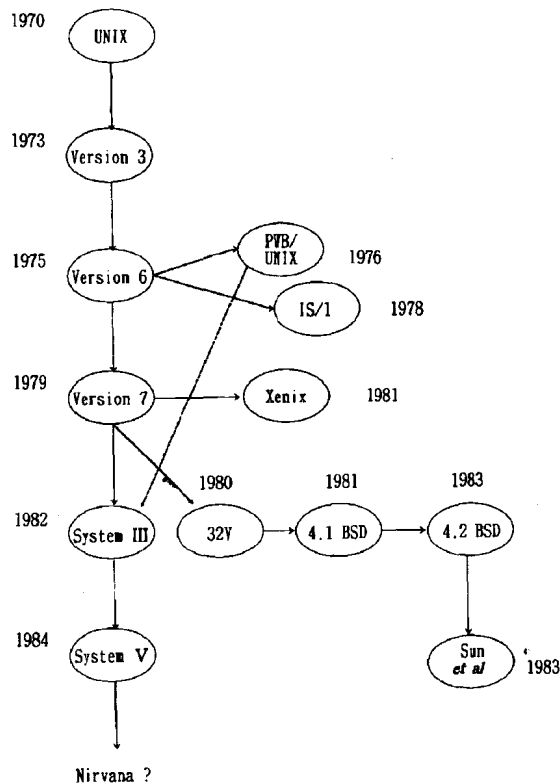


图 0-1 UNIX 系统的发展过程

系统,便不怕没有软件可供执行。

UNIX 之所以具有可移植性(即可在不同机器上执行),是因为 UNIX 几乎有 95%是用 C 语言写的,而 C 语言本身是一种较高级且与机器无关的语言。

UNIX 系统和其他操作系统一样,是电脑的控制程序,用户可把它视为一设想周到的公用程序系统(见表 0-1)。它提供了一组工具,可供用户使用或在连接后应用。

下面将介绍 UNIX 操作系统的一般和独有的特点:

1. 公用程序

UNIX 具有一系列数以百计的公用程序,这些程序可用以执行用户普遍需要的功能。譬如,Sort 公用程序可依序排列数据;而其他尚有可用以显示、打印、复制、搜寻、删除文件等等的程序。UNIX 提供了这些常用的公用程序,可省去用户自己编写程序的麻烦,从而将精力用于所计划的特定工作上。

2. 支持多用户

UNIX 操作系统是多用户操作系统,随着所用机器的不同,其可同时让多个用户在机器上执行不同的工作,而这种多用户系统允许各用户同时使用系统的资源,从而充分利用各种资源(如内存、打印机、磁盘)。

3. 支持多项工作

UNIX 系统允许用户同时执行多项工作。用户可于后台执行数项工作,而把注意力

放在终端屏幕上所显示的前台工作上。这种多任务能力可使用户提高生产力。

表 0-1 UNIX 的系统环境

1	2	3	4	5	6	7	8	9	10
Shell 和其它基本工具									
UNIX 系统内核									
计算机硬件									

- 1 邮件和消息工具
- 3 字处理程序
- 5 编辑程序
- 7 格式化程序
- 9 电子表格程序

- 2 命令解释程序
- 4 数据库管理系统
- 6 日历系统
- 8 库存控制系统
- 10 编译程序

0.1.1 Shell

Shell 是一种用户需求的公用程序。用户在终端输入的命令，交由 Shell 加以解释，而后调用适当的程序。目前有两种颇为流行的 Shell：一种是 Bourne Shell（主要用于标准的 UNIX System V），另一种是 C Shell（主要用于 BSD UNIX 系统）。

0.1.2 文件结构

UNIX 的文件结构可帮助用户建立并管理大量的文件，其结构具分级结构或树状结构。用户可在此结构中建立一主目录，以及一些必要的子目录，以便记录各文件所在的位置。目录的存在对于管理一个项目中的各文件很有帮助。除此之外，UNIX 系统允许多个用户共用一个文件，这种功能称为链接。通过链接可使一个文件出现在多个用户的目录内。

0.1.3 安全性

UNIX 系统具自有和公用文件的管理，可让用户保护自己的数据不被他人存取。另一方面，也能以简单有效的保护方式，将特定的数据或程序给某些用户共享。

0.1.4 与设备无关的输入和输出

外部设备(如打印机、终端等)对 UNIX 系统而言，就如磁盘一样，它就是一个文件。当用户下达一个命令给 UNIX 系统时，可以利用所谓“输出重定向”的功能，指定将该命令的输出送到适当的设备上。譬如，cat 公用程序通常是将某文件内容显示在终端屏幕上，当然也可将其输出重定向至某磁盘文件、打印机或是其他设备上。

0.1.5 进程(process)间的通信

UNIX 允许用户在一列命令中建立管道和过滤程序。所谓“管道”，是指某程序的输出在重定向后，成为另一程序的输入(如 `cat file | lpr`)。而所谓“过滤程序”，是指将一串输入数据处理成另一串不同数据的特定程序。过滤程序用在两个管道之间，一过滤程序在将某程序的输出处理改变后，会把处理后的结果送至另一程序做为输入。因此，我们经常结合管道和过滤程序来执行某特定工作。

UNIX 系统以其为解决标准化和可移植性问题所采用的独特发展方式、在专业团体中坚强的立足点、作为发展工具所具有的功能以及变化多端的用户接口等因素，已成为用户、软硬件制造商的选择标准。

0.1.6 数据缓存(buffering)与带外(out-of-band)数据

由字节流的服务功能来看，用户的进程能一次读写任意数目的字节，此乃无消息的边界所致。在属于字节流的服务层中(如 TCP)，可先把这些传输数据存在内部缓冲区内，然后再把这些数据往下送，以便经由网络传送到一端。数据缓冲同样也可发生在使用流量控制的通信协议中，此乃因送数据的一端比收数据的另一端速度更快，造成收的一方无法及时处理送来的数据。因此可知，一个可靠性高的通信协议通常可把数据缓冲暂存于发送端，以等待从接收端送来应答(acknowledge)上一批数据已收到的消息，如图 0-2 所示。

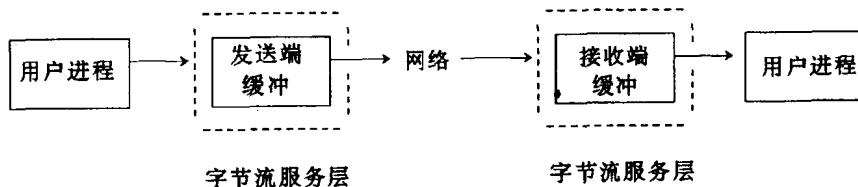


图 0-2 具有字节流的一般数据缓冲

注意：数据缓冲是可分别发生于发送和接收两端的。譬如当接收数据一端从网络接收数据的速度快于用户进程读取接收的数据时，便需在接收方采取数据缓冲的步骤。

有些时候，一个用户进程也会想把数据缓冲的功能关闭。譬如，当我们考虑终端仿真程序时，一个进程通常把每按一次键的数据(一般为一个字节长)送往另一端，此外另一端也以一次处理一个输入字符的速度进行，这样主机才可马上收到键盘所输入的每一个字，而且当键入的速度快于主机接收时，有部分字将被遗漏，不过人敲键的速度不太可能快于电脑处理的速度。

当我们采用数据缓冲之后，可能面临一个“数据堵塞”(congestion)的问题。因为有大量数据由于某种因素而留在缓冲区内，造成以后的数据无法顺利地往前送，特别是一些紧急的

消息需要赶快处理。这时便有带外数据的产生以补上述之不足。

一般带外数据也称为“加快”数据。由字义可知,字节流服务层在送此类数据时,可把它超前于已先输入的一般数据之前优先送出;同样,在接收的另一端,当有此类数据到达时,亦比早先缓冲区的一般数据优先被用户进程读取。在 UNIX 中的“signal”命令就有这种效果。由上述解释来看,带外这个名称其实也意味着用户进程使用另一不同的通道,以读取此类数据,见图 0-3。

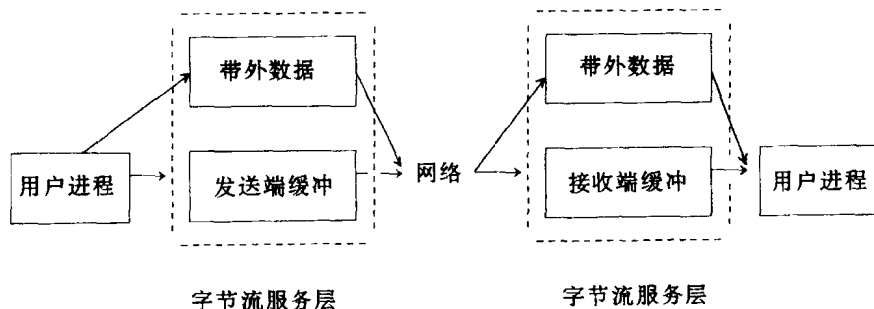


图 0-3 数据缓冲及带外数据

某些字节的层有提供带外的功能,但有些则没有。另外,如 SPP 有(为 XNS 传输层的通信协议)提供一个字节长带外数据的功能,而 TCP(为 ARPANET 传输层的通信协议)则提供无限制长度的带外数据的功能。

0.2 UNIX 网络系统的特性及主要服务功能

UNIX 网络上的第一个应用程序为 UUCP(即 Unix-to-Unix copy program)。UUCP 是从 1976 年开始开发的,而它的第一个版本是出现在 1979 年 AT&T 的 UNIX Version 7。UUCP 的主要功能是做文件传送及电子邮件。另一种应用程序 CU 也是出现在 1979 年的 UNIX Version 7,而且到目前为止仍在用。CU 主要是用在以拨号方式提供远程登录到另一电脑系统。另一种经过改良的 CU 为在 4.2 BSD 上的 tip。1980 年 9 月,美国国防部的 DARPA 机构开发一种通信协议 TCP/IP 在加州大学伯克利分校的 VAX 电脑上。在 1981 年秋季,加州大学伯克利分校推出了 UNIX 的 4.1 BSD 系统,在 1983 年 8 月又推出 4.2 BSD 系统并把 4.1 BSD 集成在其中。4.2 BSD 系统推出的同时,建立在以太网技术上的局域网也逐渐流行。这种系统允许电脑可与 ARPANET 网络连线且提供了 Socket 接口给用户开发网络应用程序。

在上述这段期间,网络由 AT&T 全力进行研究,所有研究成果并未转移给外界。但在另一方面,加州伯克利大学却把他们的网络开发成果大量地散布给外界。因此,在 1980 年便开始发展一些网络系统并把不同版本的 UNIX 系统连接起来,如 VAX 780、PDP-11/70、IBM 370 及 AT&T 3B20S。另外,AT&T 的 UNIX 系统 System III 及 System V 皆提供数据传送、

远程命令执行、远程打印及电子邮件的功能。

许多厂商在八十年代中期纷纷为 UNIX System V 发展网络套装软件,而这些软件大多支持 TCP/IP 通信协议。至于流式的输入输出系统直到 UNIX System V 的 Release 3.0 在 1986 年发表后才从贝尔实验室流传出来。此外,传输层接口(TLI)也在此版 UNIX 中提供。许多厂商以 TLI(主要用以支持 TCP/IP 协议)为基础所发展的网络套装软件陆续在 1988 年后上市。

0.2.1 层(Layering)

譬如,当我们要经由网络把一文件在两电脑之间传送时,我们通常是把这个工作分为数个部分依序分别完成。假如,原来传送文件的程序是建立在以太网的基础上,而今日却要改在专线上,则我们只要换掉有关以太网的部分而改成专线便可,至于其他部分仍可继续使用。这种分层负责不同任务的概念便称之为“层”。

0.2.2 OSI 模式

国际标准组织(ISO)的开放系统互连模式(OSI)针对网络系统提供了一项详细的标准。所以当今大部分的电脑网络皆以 OSI 模式来描述。其中,传输层尤其占有极大的重要性,此乃因其提供了其上三层正确而可信的传输数据,此外还提供了数据顺序维系、流量控制、错误诊断及数据重送等等功能。

0.2.3 一种简化的模式

在这里,我们把表 0-2 OSI 层模式中的会话层(Session)、表示层(presentation)以及应用层(application)组合成一进程层,此外亦将最下面两层组合成一称做数据链路(data-link)的单一层。这样,我们便得到一种简化的 OSI(仅具有四个层),见图 0-4。

表 0-2 OSI 七层模式

7	Application	(应用层)
6	Presentation	(表示层)
5	Session	(会话层)
4	Transport	(传输层)
3	Network	(网络层)
2	Data Link	(数据链路层)
1	Physical	(物理层)

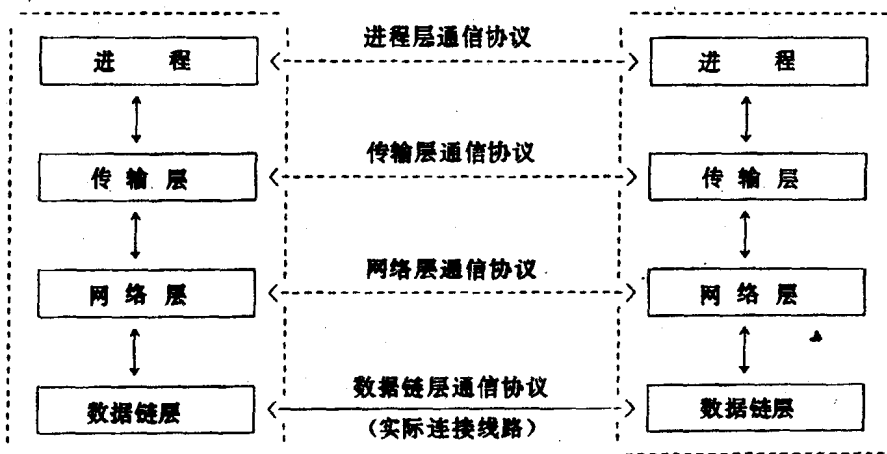


图 0-4 简化的四层模式

图 0-4 中，两个系统借着网络而连接在一起，而这实际的连接是位于两数据链路层(data-link)层之间，并以实线表示。实际数据的传输是某一系统的进程层往下送，并经传输层、网络层及数据链路层而到达实际的网络接线，然后再由此线到另一系统并往上经数据链路层、网络层而到达进程层。虽然由上可知实际的连线是建立在最低的一层之间，但是所谓的通信协议却存在两系统的水平相对两层之间，这种关系被称为“对等”(peer-to-peer)通信协议，在图 0-4 中以四条水平线表示。

由于数据在相邻的上下两层间快速上下流动，所以相邻层间的接口是网络系统程序所关注的。譬如加州伯克利大学针对传输层与上方相邻的另一层的接口提出“Socket”，而 AT&T 亦提出 TLI。

0.2.4 客户-服务器(client-server)模式

从应用程序的观点来看(非通信接口程序的观点)，一般皆采用此种模式来探讨网络上不同主机的关系。通常身为服务器(server)的主机都有一进程在等待由身为客户的主机送来要求服务的消息，以便提供不同的服务功能。两主机间的消息来往及处理过程如下所示：

1. 服务器上的进程在启动之后便经过一些初始化的操作，然后便进入“等待”的状态，以等候要求服务的消息由客户机上的进程送来。
2. 客户机借着网络与主服务器连接起来。而其上的进程反复地接收用户送入的命令，并把此命令转成一要求服务的消息送给服务器。
3. 当服务器上的进程收到要求，然后服务完毕之后，便再度进入“等待”状态以等候下一个要求的到来。

0.2.5 网络上的主要服务功能

1. 远程过程调用(RPC)的功能由程序库提供，此程序库中的过程使得一调用进程可借

由它们调用另一服务进程上的过程去执行,其中调用进程与服务进程并不一定皆需在同一主机中。

2. 外部数据表示(XDR)是一种可移植性的数据格式表示法。RPC 使用 XDR,以确保在不同的电脑、操作系统及电脑语言中,所有数据表示的意义是相同的。
3. 网络文件系统(NFS)是一种允许用户把位于不同文件系统中的目录经由网络而安装在某一文件系统之中,这样用户利用某一文件系统抓取远方的文件时,就如同抓取本地文件一样便利。NFS 的目的在于使网络上的工作站能使用存于网络上任何文件系统上的文件(如图 0-5)。

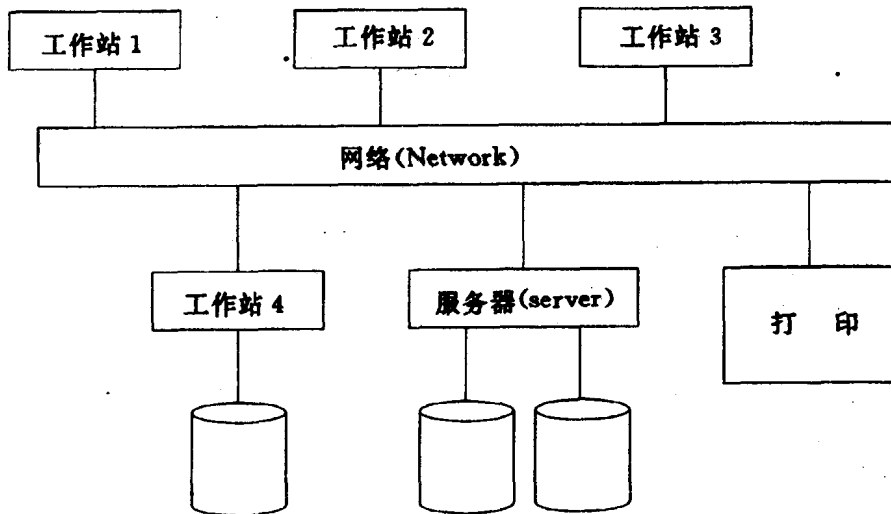


图 0-5 工作站与服务器的关系

假设你正使用的主机名称为“client”,而想读一些存于另一部主机(称为 docserv)的文件,可使用下列命令把文件所在的目录安装到你的文件系统中:

```
client # /usr/etc/mount docserv:/usr/man/usr/man
```

接着可使用“mount-P”命令看到所有目录安装的情况如下:

server:/roots/client	/	nfs rw,hard	0 0
server:/usr	/usr	nfs ro	0 0
server:/home/server	/home/server	nfs rw,bg	0 0
server:/usr/local	/usr/local	nfs ro,soft,bg	0 0
docserv:/usr/man	/usr/man	nfs ro,soft,bg	0 0

你不仅可安装目录,更可直接安装文件,图 0-6 表示不同文件系统中有关安装的关系。其中椭圆形代表不同的主机,长方形代表经由 NFS 而被安装上的目录或文件。在图上共有五个长方形在主机 client 下,此即为表示 client 之下有五个目录或文件是安装上

去的,例如/usr/man 目录其实是主机 docserv 上的目录/usr/man(可以 docserv : /usr/man 表示),经由 NFS 而安装在 client 上。

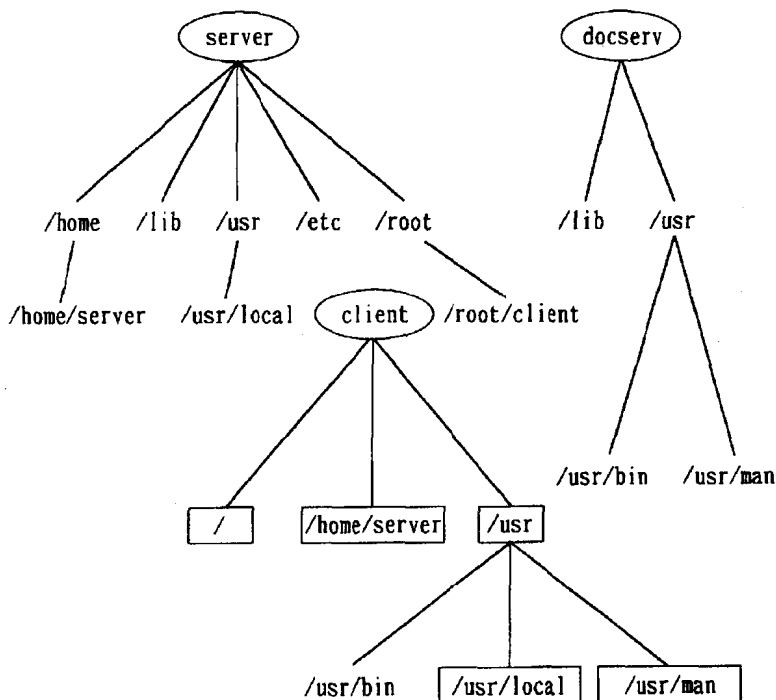


图 0-6 网络文件结构与关系

4. 远程文件共享(RFS)提供一种将远方的数据视为存于自己机器上数据的方式。像 NFS 一样,RFS 也允许应用程序经由网络系统把共享远程数据的操作简化为如同抓取自己机器上的数据。在 NFS 中,事务处理之间是彼此独立的,所以当其他服务器或客户(client)发生故障时,本身的系统内并不需要做复原的工作。然而 RFS 中保存着一些跨越不同事务处理间的状态数据。因此当有其他服务器或客户发生故障时,就需做一些复原的工作。RFS 在使用上极似 NFS,而其不同之处将会在稍后的章节中介绍。
5. 端口映射(Portmapper)是一种系统服务功能,它提供一种记录器用以追踪在每一机器上不同的服务功能所相对应的端口(即逻辑通道)。这样,任何 RPC 应用程序在寻找需要的服务功能时,便可经由这些端口号码找寻所需的服务功能。

0.3 流(STREAMS)

当我们设计不同的系统接口程序时,常常会发现部分相同的功能出现在不同的接口程序中。譬如在设计网络通信协议的界面程序时,基本上包含有硬件接口控制部分及通信协议部分,其中通信协议部分对大多数网络程序而言是相似的。为避免相同功能的重复设计,模