

UNIX

教程

网络篇

Unix Networking Clearly Explained

(美) Richard L. Petersen 著

蒋文保 高书杰 刘宝旭 等译



机械工业出版社
China Machine Press



MORGAN
KAUFMANN

UNIX教程

网络篇

(美) Richard L.Petersen 著

蒋文保 高书杰 刘宝旭 等译



机械工业出版社
China Machine Press

本书详尽地讲述了用于网络及Internet的各种UNIX工具。

本书为每个邮件程序、新闻阅读器和Internet应用的命令、选项及特性提供了简捷的图表示意,使读者非常易于深入掌握各种Unix网络应用技术。作者对主要的Unix网络应用进行了透彻的描述,使人们能较好地掌握Unix网络的各方面知识。

本书可作为Unix用户学习的教程和参考书。

Richard L. Petersen: Unix Networking Clearly Explained

Original edition copyright©1999 by Academic Press. All rights reserved.

本书中文简体字版由美国Morgan Kaufmann公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号:图字:01-1999-3677

图书在版编目(CIP)数据

UNIX教程——网络篇 / (美)彼德森(Petersen, R.L.)著;蒋文保等译. —北京:机械工业出版社, 2000.6

书名原文: Unix Networking Clearly Explained

ISBN 7-111-08063-7

I. U… II. ①彼… ②蒋… III. UNIX操作系统-教材 IV. TP316.81

中国版本图书馆CIP数据核字(2000)第24739号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码100037)

责任编辑:李云静

北京昌平奔腾印刷厂印刷 新华书店北京发行所发行

2000年6月第1版第1次印刷

787mm×1092mm 1/16·15.25印张

印数: 0 001 - 6 000册

定价: 25.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译 者 序

UNIX是一种适用于多种硬件平台的多用户、多任务操作系统、最初的UNIX操作系统是1969年由美国AT&T公司贝尔实验室开发出来的。从1969年至今，它经历了一个从开发、发展、不断演变和获得广泛应用，以致逐渐成为网络服务器和工作站的最重要操作系统平台的过程。UNIX是最灵活而又最难掌握的操作系统之一，同时它也是一个具有非常强大生命力的系统。另外，UNIX系统具有很强的网络特性，当今许多用于网络通信的工具，包括著名的TCP/IP协议，都是首先在UNIX系统上设计出来的。

本书详尽地讲述了用于网络及Internet的各种UNIX工具，着重介绍客户端的应用软件，包括各种UNIX邮件程序、通信工具、新闻阅读器、Internet访问工具以及远程访问操作。书中针对每一种UNIX网络工具的使用，都提供了大量的示例和简捷的图表。

本书通俗易懂，内容充实，实用性强，非常有助于读者学习各种UNIX网络应用知识。

本书主要由蒋文保、高书杰、刘宝旭翻译，并由蒋文保统稿和审校，参加本书翻译、录入工作的人员还有张晓梅、钱桂琼等。

译者水平有限，不当之处敬请读者批评指正。

蒋文保

2000.4

前 言

采用网络技术可以将不同的Unix系统连接起来。一个网络就是一组通过通信线路连接在一起的计算机，它可以连接多台计算机，可以连接本地的机器，也可以连接全球范围的系统。在Unix网络中，每台计算机一般都有自己的Unix系统，不同的Unix系统之间通过网络进行通信。通过网络，用户可以将消息传递到其他Unix系统上的用户，还能在两个系统之间传送文件，并能发布在其他系统上的用户也可读取的新闻和讨论话题；同时，还能查找另一个系统上的程序、文章等信息，并可将这些信息传送到自己的系统上。

本书介绍4种不同的Unix网络应用技术：邮件程序（mailer）、新闻阅读器（Newreader）、网络及Internet工具、远程访问命令。邮件程序用于传送和接收消息；新闻阅读器用来阅读和发布Usenet新闻信息；网络及Internet工具是Unix提供的用于访问网络资源的一套工具，例如提供远程登录功能的Telnet、用于文件传送的FTP、提供友好易用界面的Gopher菜单和Web浏览器；远程访问命令能让你在自己的系统上执行远程系统上的命令。

本书着重向用户介绍网络的使用技术，而不是网络管理技术。它主要针对网络环境下的一般Unix用户，介绍Unix系统上的各种网络应用。

客户/服务器

在网络技术发展以前，大部分计算任务都是在带有许多终端的大型机上完成的，用户必须与大型机直接相连的终端上组织命令。随着个人计算机（PC）的发展，人们开始只让大型机完成一些难度大的计算任务，然后将计算结果下载到他们自己的PC上，以进行另外一些特定的处理，例如将有关数据集成到采用字处理器或电子表格进行处理的报告中。

随着局域网和小型机的发展，几台PC和小型机可以连接在一个网络上，小型机可以替PC完成一些密集型的计算任务。这种情况下，用户可以直接向相连的小型机发出任务请求，小型机完成计算任务后，通过网络将结果自动送回到提出请求的PC上。这样，就不用像以前那样，需专门登录到一台大型机上才能完成计算任务。在这样的网络中，发出请求的PC就成为客户，而接收请求并完成计算任务的计算机就是服务器。在这种客户/服务器关系中，客户向服务器发出请求，服务器响应请求并完成计算任务，然后将计算结果通过网络送回到客户端。后来，工作站取代了大部分小型机充当网络中的服务器，这种工作站既有PC的灵活特点，又具有小型机的计算能力。

Unix操作系统是许多客户/服务器网络所选用的系统。由于具备多用户、多任务的特性，因此Unix系统可在同一时间里接收和处理许多不同客户的请求。Unix系统的可扩充性满足网络需求，它集成了文件共享和识别其他连接系统的网络功能。另外，许多最初的用于网络通信的工具，例如提供文件传输功能的FTP，也是首先在Unix系统上开发的。

服务器是一个系统或一个系统的一部分，用以完成网络上其他用户所请求的任务。它专门用来完成如打印文档或访问数据库之类的任务。它能接收网络中一个系统发出的特定任务请求，这种请求是由该系统中一个用户发出的。服务器完成任务后，就返回结果给提出请求

的系统。例如，当打印完一个文件后，服务器就给出一个响应，告诉用户文件已经打印完毕。在访问数据库的情况下，它就送回数据记录。这种体系结构可以避免大量的资源浪费，例如，在整个网络中可以只给服务器配置一台打印机；而不用为每台计算机都配置它自己的打印机，就能完成所有的打印任务。网络上的其他系统要打印文件时，只需简单地传送文件给服务器或告诉它文件所在位置，就可以请求服务器打印了。同样，在数据库应用中，你不用在每个系统上都创建一个同样的数据库；而只需在一台服务器上建立一个数据库。其他系统需查询或修改数据库时，只要给服务器发送相应的请求就行了。

客户是向服务器发出请求的一些系统或应用。最初“客户”和“服务器”这些术语都是针对网络上的硬件而言的，一般指发请求的计算机或接受并处理请求的计算机。后来，这些术语也用来指系统上一些特定的软件程序。实际上请求和处理请求都是由程序进行的。运行在某台计算机上的Unix操作系统可以用作一个服务器，接受请求并运行几个程序处理请求。而这些请求也是由运行在某台计算机上的客户程序发出的。例如，一个邮件客户就是一个邮件程序，供用户撰写消息并发送给邮件服务器，请求服务器将消息发送到网络上的另一指定用户；邮件服务器是一个由服务器上的操作系统执行的服务程序，它可以找到网络上的用户，并能发送消息。另一个典型的例子是Web浏览器和Web服务器。Web浏览器是一个客户程序，它向运行在某个站点上的Web服务器发送访问Web页面的请求；Web服务器是一个服务程序，能监听到浏览器的请求，并将相关的Web页面发送给浏览器。

因此，一个服务器实际上并不一定占用整台计算机或整个操作系统，它可能就是仅占用系统一部分的一道程序。例如，在一个Unix系统上可以同时实现一个Web服务器、一个Gopher服务器和一个FTP服务器。它们的服务程序可以同时运行，Web服务器程序提供客户访问该Unix文件系统下的Web页面文件，Gopher服务器提供客户访问Gopher目录下的文件，而FTP服务器则为公用访问文件服务。服务器软件的实现和维护主要是网络管理的任务，不在此书讨论范围，我们这里主要讲述网络环境下用户所使用的一些客户程序。

TCP/IP和UUCP

网络是根据通信协议连接各个系统的。Unix系统可以使用两种不同的网络协议：一种是TCP/IP (Transmission Control Protocol/Internet Protocol, 传输控制协议/网际协议)，另一种是UUCP (Unix-to-Unix communications Protocol, Unix与Unix之间的通信协议)。用于Internet上的TCP/IP协议也可以用在局域网上。采用TCP/IP的网络通常具有专用的连接，例如以太网连接和远程拨号连接。大多数Unix系统都采用TCP/IP协议接入网络。

TCP/IP协议族包含了许多种协议，每种协议在TCP/IP网络体系结构中都有着自己的特定用途。其中两种基本的协议是TCP（传输控制协议）和IP（网际协议），其他一些协议可提供各种网络服务，例如，DNS (Domain Name Service, 域名服务) 提供地址解析服务，FTP (File Transmission Protocol, 文件传输协议) 提供文件传输服务，NFS (Network file System, 网络文件系统) 提供访问远程文件系统服务。

TCP/IP协议并不限于Unix系统，它适用于任何一种操作系统，包括小型机上的VMS、大型机上的IBM系统和PC上的Windows系统，这也是Internet采用TCP/IP协议的一个理由。不过，别忘了TCP/IP协议最先是在Unix系统上开发出来的，并用于大多数Unix网络中。

UUCP是一种可选的协议，仅适用于Unix系统之间的网络通信。它是一种比较早的协议，

针对还没有连上网络的各个系统之间的互操作而设计出来的。采用了UUCP，在预定的时间通过电话线，一个系统就可以与另一个系统相连，一次可处理一批通信任务。UUCP特别适用于这样的应用：与某一个特定的系统直接建立连接，然后传递数据，最后断开连接。

电子邮件

Unix系统具备电子邮件功能，可供你在网络上与其他用户交换消息。你可以将消息发送给同一系统上的任何用户、或者同一网络中的任何系统，甚至Internet上的任何用户。在Internet这类网络中，每个用户都有一个电子邮件地址，该地址包括用户名和用户所在系统的地址。将网络地址添加到系统名中就形成了一个完整的域名，它可以用来定位网络中的任何一个系统。使用电子邮件工具`mailer`，就可以在Unix系统上收发消息。大多数Unix系统都配备了几个通用的邮件程序，尽管它们都有相同的基本功能，但各自提供的界面却差异很大。本书共讨论了四种不同的邮件程序：`Mailx`、`Elm`、`MH`和`Pine`。其中每种都具有自己独特的界面，并且都是功能强大的邮件程序，具备许多增强的特性。

`Mailx`已成了标准的组件，几乎所有的Unix系统都选用它。`Mailx`提供的是命令行界面，虽然称不上易于使用，但却有很强的功能。`Elm`提供一个基于光标的全屏界面，大大简化了消息的处理，只要移动光标以及输入单个字母命令就能完成工作。`MH`却提供另一种不同的方法，它将电子邮件操作集成到标准的Unix命令行界面中，即可以使用Unix命令行执行邮件操作，而不用启动一个单独的执行环境；这样，采用简单的Unix命令就能检查邮件或发送消息。

`Pine`是一个颇受欢迎的邮件程序，已在许多不同的操作系统上得到实现，包括Windows、Macintosh和Unix等不同版本的`Pine`。它采用MIME协议，可以附加二进制文件到消息中。使用`Pine`就能随消息一起发送如图像这样的二进制文件，然后让另一个也使用`Pine`的用户接收并打开它。同时，`Pine`还具有Usenet新闻功能，可以用作一个newsreader。

这里描述的邮件程序都不是基于X-Window的，许多X-Window GUI界面（例如Motif和Open-look）提供了易于使用的X-Window邮件程序。另外，像Netscape Communicator和Mosaic这样的Web浏览器也提供基于X-Window的邮件程序。虽然这些易于使用的邮件程序都可以用来收发消息，但可能不像`MH`或`Pine`这类基于光标或逐行方式的邮件程序那样具有许多增强功能。

本书还将讨论Unix支持的其他一些通信方式。例如，使用`writer`命令向系统上的另一个用户立即发送消息。使用`talk`命令可与另一个用户建立双向通信，即双方在拆分的屏幕上可以同时输入消息，并同时显示在双方的终端上。

新闻阅读器

新闻阅读器用于阅读Usenet新闻文章，而所有这些文章都放在可供你订阅的新闻组（newsgroup）里。新闻阅读器从你所订阅的新闻组中下载并显示文章。新闻阅读器的种类很多，大多数像Netscape Communicator和Mosaic这样的Web浏览器也都有新闻阅读器。本书将讨论Unix系统上的一些基于光标的新闻阅读器：`rn`、`trn`、`tin`和`nn`。

`rn`是最先开发出来的新闻阅读器之一，它提供一种简单的逐行方式的界面，但包含了强大的搜索和选择功能。`rn`采用页面形式显示文章，就像`more`或`pg`命令那样。`trn`是一种线索化的新闻阅读器，采用全屏显示供你挑选文章，所有与某个话题相关的文章都归并到一个线索

上。使用基于光标的线索树，可以方便地从一篇响应的文章跳到另一篇相同话题的文章。

tin能为新闻组列表和文章列表提供全屏的基于光标的界面。跟trn一样，它也能支持线索，列出相关话题的文章。tin可提供一套扩充的特性和功能。nn是与tin类似的一种全屏基于光标的新闻阅读器，但是不具备扩充功能。

网络和Internet资源

在Unix系统中，大多开发出来的客户/服务器软件，都是用来存取网络上各种各样资源的。资源可能是一个文件、到另一个系统的远程登录、网络文件的位置或者是列出其他资源的一些菜单和页面。提供访问这些资源的软件相当于提供一种服务。用户使用客户软件发出服务请求，然后由相应的服务器软件接受并完成任务。Unix系统提供的网络服务有Telnet、FTP、Archie、WAIS、Gopher和Web。本书将讨论客户端应用，而服务器端的操作是网络管理的话题。

Telnet可用来连接到远程系统上，就像使用一个终端那样登录到系统的一个帐户上。采用FTP，可以与远程系统建立连接、登录、传送和接受文件。虽然可以使用FTP连接到任何支持FTP的系统上，但它主要用于从FTP站点上下载文档和软件包。Archie用于搜索目录和文件，并返回它们所在FTP站点的列表。可以使用通常的表达模式定位查找文件所在的站点，然后使用FTP下载文件。WAIS (Wide Area Information Service, 广域信息服务) 是另一种查找服务，可以定位FTP、Gopher或Web站点上的任意一种文档。它通过添加文档索引，使文档更易于查找。Gopher为用户访问网络资源提供了一种方便的途径，它具有一种基于菜单的非常易用的界面。使用Gopher客户软件连上Gopher站点后，就可以看到Gopher菜单，它列出了该站点上的资源和一些链接到其他站点上的资源链接。一个Gopher菜单可以链接到其他的Gopher菜单、或者别的FTP站点上的网络资源。你可以方便地查找Gopher菜单条，并在另一个Gopher菜单里显示出所有匹配的结果。

Web或WWW是一种由采用超文本界面的Internet站点组成的网络，它为用户提供了访问全球Internet资源的一种方便途径。超文本文档可用来显示图像、表格以及格式化的文本，在WWW上的超文本文档被称为Web页面。在一个Web页面中可以包含其他Internet资源的地址，这种嵌入地址就叫做超链接或链接。通过带链接的Web页面，可以访问FTP或Gopher站点上的Internet资源、Telnet连接以及其他的Web页面。

用来访问和显示Web页面的客户端软件就叫做Web浏览器。通过Web浏览器，可以显示不同站点的Web页面。Unix系统上具有好几种浏览器供用户选择，其中Netscape Communicator和Mosaic都是基于X-Window的浏览器，它们带有按钮、菜单和窗口，能显示图片和运行Java小应用程序 (applet)。Lynx是一种基于光标的浏览器，不能运行在X-Window上，也不能显示图片。它实际上是一种基于文本的浏览器，提供一种类似于Pine那样的界面。

远程访问

远程命令用于完成网络中远程系统上的操作。这些命令都是在Unix命令行中执行，就像其他Unix命令一样。对于TCP/IP网络上的系统，可以使用远程命令完成诸如登录到另一个系统、远程拷贝文件等操作。通用命令名称前加上字母“r”就表示远程操作命令，例如rcp就是用于从一个系统到另一个系统远程拷贝文件的命令。使用rlogin命令，可以登录到另外一个系

统的帐户上，很像Telnet的功能。通过rsh命令，就能执行远程系统上的命令。有些命令可用于检查系统的状态，例如ping命令用于测试系统是否正确连接到网络上。这些远程命令都具有实时操作的优点，例如用远程命令可以立即完成在远程系统上拷贝文件和执行命令的操作。不过，请记住这些命令只能用在采用TCP/IP协议的网络系统。

UUCP网络提供另一套不同的远程命令集，用于完成远程系统上的操作。例如，使用uucp命令，可以将文件从一个系统传送到另一个系统；uuname命令用于显示所有可访问到的采用UUCP连接的系统；uux用来执行远程系统上的命令。许多UUCP命令都相应于TCP/IP的远程访问命令，例如，uucp就像rcp，uux类似于rsh。

在UUCP网络中，远程命令按成批方式发送到其他系统上，数据传送一般通过调制解调器和标准的通信线路完成。远程系统执行接受到的命令，然后送回结果。例如，远程拷贝文件的请求命令与其他命令一起成批发送到远程系统上，执行完毕后，就在下一步数据传送中将已拷贝的文件送回到你的系统上。

如何使用这本书

本书分成四个部分讨论不同种类的Unix网络应用。第一部分讨论了邮件程序；第二部分讨论用于检索和显示Usenet新闻的新闻阅读器；第三部分讨论用来访问网络资源的一些工具，包括Telnet、Gopher菜单、FTP站点、Web浏览器和Web页面；第四部分讨论远程访问命令。

每一部分的第一章中都列出了该部分所需的一些基本信息，例如，在第三部分的第一章（即第9章）里，讨论了后续章节内容涉及到的网络寻址概念。

原书书号：ISBN0-12-552145-6

原出版社网址：<http://www.mkp.com>

目 录

译者序
前言

第一部分 电子邮件

第1章 电子邮件概述	1	件	24
1.1 用户网络地址	1	2.3.1 mailx别名	24
1.1.1 Internet域寻址方式	2	2.3.2 mailx选项	24
1.1.2 UUCP路径寻址方式	3	2.3.3 管理邮箱文件	26
1.2 发送和接收邮件	3	第3章 电子邮件程序Elm	28
1.3 收到邮件的通知	4	3.1 使用Elm发送邮件	28
1.4 自动接收邮件	5	3.2 使用Elm接收信件	30
1.5 二进制邮件和档案文件	5	3.3 退出Elm程序	32
1.5.1 编码二进制	6	3.4 删除信件与恢复删除信件	32
1.5.2 MIME	7	3.5 回复信件	33
1.6 在线通信:write和talk	7	3.6 发送新信件	33
1.6.1 直接连接: write	8	3.7 保存信件	33
1.6.2 交互通信:talk	9	3.8 读取邮箱文件	34
第2章 电子邮件程序mailx	10	3.9 Elm别名	35
2.1 发送邮件	10	3.9.1 别名菜单	35
2.1.1 标准输入及重定向	10	3.9.2 .elm和aliases.text	36
2.1.2 给多个用户发送同一封信件	11	3.10 Elm选项	36
2.1.3 拷贝信件内容到一个文件里	12	3.11 Elm命令小结	37
2.1.4 采用tilde命令编辑信件	12	第4章 电子邮件程序MH	39
2.2 接收信件	15	4.1 使用MH发送邮件	39
2.2.1 信件列表符及当前信件标记 符	16	4.2 使用MH接收信件	40
2.2.2 显示信件	17	4.3 使用MH显示信件	41
2.2.3 信件的删除及恢复删除	18	4.4 指定MH信件	41
2.2.4 回复信件和发送新信件	19	4.5 打印、保存和回复信件	42
2.2.5 退出邮件shell	20	4.6 删除信件	43
2.2.6 在邮箱文件里保存和访问 信件	21	4.7 选择信件	43
2.2.7 在文件里保存信件内容	23	4.8 使用信件夹	45
2.3 mailx别名、选项及邮件初始化文 件	24	4.9 在邮件脚本里使用MH	46
		4.10 MH命令小结	46
		第5章 Pine	48
		5.1 撰写和发送邮件	49
		5.1.1 填写信件头	50
		5.1.2 编辑信件正文	51
		5.1.3 Pine地址簿	52

5.2 接收信件	53	7.14 小结	104
5.3 选择信件夹	56	第8章 tin和nn 新闻阅读器	105
5.3.1 信夹集合	57	8.1 tin 新闻阅读器	105
5.3.2 新闻组	58	8.2 tin 新闻组选择屏幕	105
5.4 Pine附件	59	8.3 新闻组索引屏幕	108
5.5 Pine配置	61	8.4 tin 线索列表	111
5.6 Pico 和Pilot	65	8.5 tin 文章阅读器	112
第二部分 新闻阅读器		8.6 tin 文章过滤	114
第6章 Usenet及新闻阅读器: rn 和 readnews	67	8.7 tin 命令行编辑和历史	116
6.1 发布文章: Pnews	68	8.8 tin 配置	116
6.2 .signature 文件	70	8.8.1 tin 文件	117
6.3 新闻阅读器rn	70	8.8.2 tin 选项	117
6.3.1 新闻组列表	70	8.8.3 tin 配置变量tinrc	118
6.3.2 文章列表	72	8.8.4 环境变量	122
6.3.3 显示文章	74	8.8.5 新闻组属性	123
6.3.4 保存文章	74	8.9 nn新闻阅读器	125
6.3.5 标记文章	76	第三部分 网络及Internet工具	
6.3.6 文章选择	76	第9章 TCP/IP 网络互联技术	129
6.3.7 答复文章: 跟随文章和信件	77	9.1 TCP/IP 协议	129
6.3.8 rn选项	79	9.2 域名	130
6.4 新闻阅读器readnews	81	9.3 Internet和Intranet	131
6.5 小结	83	9.4 TCP/IP 地址	132
第7章 新闻阅读器trn	84	9.5 TCP/IP 网络地址	133
7.1 trn的新闻组界面	84	9.6 TCP/IP 配置文件	134
7.2 trn选择器	86	9.6.1 标识主机名文件: /etc/hosts	134
7.3 选择器显示模式: 文章、主题和 线索	89	9.6.2 网络名文件: /etc/networks	135
7.4 查找文章	91	9.6.3 主机名文件: /etc/hostname	135
7.5 显示文章	92	9.6.4 域名服务 (DNS) 文件: /etc/ host.conf和/etc/resolv.conf	136
7.6 标记文章	94	第10章 Telnet	137
7.7 回复文章	95	10.1 Telnet连接	137
7.8 保存文章	95	10.2 公共服务	138
7.9 trn线索树	96	10.3 Telnet命令	139
7.10 trn选项	99	10.4 Telnet配置	142
7.11 查补代码	100	10.5 Telnet的.telnetrc文件	143
7.12 trn环境变量	102	第11章 FTP	144
7.13 trn宏	103	11.1 FTP连接	144
		11.2 FTP目录和文件命令	145

11.3 文件传输	148	14.3 Web浏览器	193
11.4 代理FTP	152	14.3.1 Netscape Communicator	193
11.5 匿名FTP	152	14.3.2 Mosaic	195
11.6 自动登录	155	14.3.3 Lynx	195
11.7 FTP宏	156	14.4 Java	197
11.8 文件名的映射和转换	157	14.5 超文本标记语言HTML	197
11.9 FTP传输参数	158	14.5.1 标题和列表	200
11.10 tar档案文件	159	14.5.2 访问Internet资源	202
11.11 压缩文件	160	14.5.3 Web页面示例	203
11.11.1 compress、uncompress和 zcat	160	14.5.4 图像和声音	205
11.11.2 gzip、gunzip和gzcat	161	14.5.5 表单和公共网关接口	207
11.11.3 zip和unzip	161		
第12章 Archie和WAIS	164	第四部分 远程访问	
12.1 Archie	164	第15章 TCP/IP远程访问操作	211
12.1.1 Archie客户软件	164	15.1 网络信息检测命令: rwho、 ruptime和ping	212
12.1.2 Archie服务器	166	15.2 远程访问权限文件: .rhosts	212
12.2 WAIS	170	15.3 远程登录命令: rlogin	213
12.2.1 freeWAIS	171	15.4 远程文件拷贝命令: rcp	213
12.2.2 WAIS服务器	171	15.5 远程执行命令: rsh	214
第13章 Gopher	176	第16章 UUCP	216
13.1 Gopher菜单	176	16.1 UUCP的寻址方式	216
13.2 Gopher书签	178	16.2 显示连接系统命令: uuname	218
13.3 Veronica	180	16.3 UUCP连接管理命令: uucico 和uuxqt	218
13.4 Gopher配置文件	181	16.4 UUCP命令	219
13.5 Gopher服务器	183	16.5 文件传输命令: uuto和uupick	219
13.5.1 Gopher目录	184	16.6 文件拷贝命令: uucp和uustat	221
13.5.2 Gopher索引	186	16.7 远程执行命令: uux	222
13.5.3 一个Gopher站点的例子	186	16.8 远程登录命令: cu和ct	223
第14章 World Wide Web	189	Unix术语	226
14.1 URL地址	189		
14.2 Web页面	192		

第一部分 电子邮件

第1章 电子邮件概述

Unix系统上具有电子邮件程序，可以用来发送和接收信件。通过它，你可以发信件到本地系统上的任何用户、另一个与你网络连接的系统上或到Internet上的任何人。在你的系统上，每个用户有一个地址，该地址同他的登录名相同。要发信件到另一个用户，你仅需要知道此用户的登录名。对于其他想发信件到你地址的用户，他们需要知道你的地址，也就是你的登录名。如果需要发信件到其他系统的用户，你需要知道他们的网络地址。它通常由用户登录名、系统的名字和系统的位置组成。假如知道用户的地址，无论用户在那里，你都可以发信件到此用户。

在Unix系统中，使用电子邮件程序，你可以发送和接收信件。在大多数系统中，有几个通用的邮件程序可以用。尽管它们从事同样的接收和发送信件的基本任务，但却有不同的接口。根据所使用的邮件程序，你可以用不同的方式接收和发送信件。本书讲述四种不同的邮件程序:mailx、Elm、MH、Pine。每一种有不同类型的接口。mailx使用一个简单命令行接口，提供自己的shell操作。大部分Unix系统使用它，它被作为一个标准。Elm有一个全屏幕接口，并利用单键命令，如同Vi编辑器中的命令。MH定义了一个Unix命令集，在用户的shell中可以直接发送和接收信件，而不用在特殊邮件程序中操作。Pine 同样使用一个全屏幕接口，而且集成了如新闻阅读器那样的功能。

假如二进制文件当作文本文件编码，就可以通过Unix电子邮件发送二进制文件，然后，接收到的信件可以被解码为二进制文件。此外，通过Unix应用程序write 和talk，可以在本地系统上的用户之间接收和发送实时信件。它们不是邮件程序，它们在用户之间直接连接，如同收音机和电话一样。

本章讨论网络地址、使用mailx进行发送和接收信件的基本电子邮件操作、发送二进制文件的方法、在登录的用户之间实时通信的使用工具。在下面章节里，将深入地讨论不同的电子邮件程序。

1.1 用户网络地址

在Unix系统中，用户有一个自己的电子邮件地址。要发信件到另一个用户，需要使用那个用户的邮件地址。当前，大部分用户使用Internet地址连接到网络，每一个用户的邮件地址由用户名和它们的系统Internet地址组成。对于在violet.berkeley.edu系统上的justin用户，他的地址将是justin@violet.berkeley.edu。UNIX支持地址替换方法。在相同系统上的用户之间，只需要使用对方的用户名(登录名)做地址。在同一个系统的其他用户要向用户justin发送信件时，仅需要使用justin做地址。然而，当发信件到其他系统的用户时，你不仅需要知道他们的用户名，而且还要知道他们系统的地址。Unix同样支持使用UUCP表达地址的较老方法。它通常用

于大多数早期的网络中,在这种网络上,通过网络连接或拨号modem连接,信件从一个系统传送到另一个系统。

对于不同系统上需要彼此通信的用户,他们的系统需要连接到一个网络上。这些系统间接地彼此通过网络相连,一个系统被连到其他系统,此系统又连到另一个系统,依此类推。你可以通过中间连接的系统发送信件到达一个网络的远端。例如,如果violet系统连到了stan系统,它又连到了bell系统,那么通过stan,在violet系统的用户可以到达bell系统的用户。信件作为批量处理的一部分,从一个系统发送到另一个系统,到达它们的目标系统。

网络上的地址要求系统地址独一无二地被标识。每一个地址有自己的名字,还有用于说明系统位置的名字或系统拥有者的名字。然后你可以使用这样一个完全的网络地址,发邮件到全球网络上的任何一个系统。在Internet上,你可以把信件发送到世界上的任何系统。

有两种不同的寻址方式:域和路径。域寻址方式来源于Arpanet,就是现在的Internet前身。域寻址方式广泛地应用于Internet。对于Unix System V,AT&T开发了另一个网络通信包,称为Unix-to-Unix 通信协议(UUCP),这种UUCP网络通常使用路径寻址方式。尽管域寻址方式已成为Internet和Intranet的标准寻址方式,但这两种寻址方式可以混合。

Internet域寻址方式使用每个系统独一无二的地址,处理通过网络的信件。路径寻址方式,说明在你和你想去连接的系统之间的所有中间系统。这使得路径地址比域地址复杂得多。在表1-1中说明了这两种寻址方式。

表1-1 网络邮件地址:域和路径

地 址 格 式	作 用
login-name@system.domain	域邮件地址(Internet),如: chirs@violet.rose.edu
system!login-name	路径邮件地址,用在System V(UUCP)中,如: rose!violet!justin
system\\login-name	路径邮件地址用在C-Shell(UUCP)中,如: rose\\!violet!justin

1.1.1 Internet域寻址方式

域寻址方式的设计目的,是为了简化另一个系统上某个用户的定位。域寻址方式赋予系统一个域地址,它给系统一个独一无二的地址。一旦用户指定了这个独一无二的地址,网络就可以跟踪到。

域名地址由标识系统的主机名、标识网络的域名和标识网络类型的扩展名组成。每一个部分由点号分开。下面是域名地址的语法格式:

主机名.域名.扩展名

在特定系统上的用户登录名,加上域地址就形成完全的Internet邮件地址,其格式如下:

登录名@主机名.域名.扩展名

要发信件到violet.berkeley.edu系统上的justin,只要按如下形式给出地址即可:justin@violet.berkeley.edu。在下面例子给出的域名地址中,主机为violet,网络为berkeley,它的类型为教育机构,因此扩展名为edu:

```
$ mailx justin@violet.berkeley.edu < mydata
```

1.1.2 UUCP路径寻址方式

路径寻址方式仅使用系统地址，不用域地址。系统地址放在用户名前，用感叹号分开。下面是路径寻址的语法表示：

```
system!login-name
```

在下一个例子里，通过mailx发信件到一个名为violet的Unix系统上的用户justin：

```
$ mailx violet!justin < mydata
```

在C-shell里，路径地址寻址方式要求一个反斜杠放在感叹号的前面。在C-shell中感叹号本身代表历史(history)命令。反斜杠将转义感叹号为感叹号字符，而不是作为history命令。下面是C-shell路径地址的语法格式，以及C-shell路径用在mailx命令中的例子：

```
system\!login-name
```

```
% mailx violet\!justin < mydata
```

在使用路径寻址方式表示的网络中，在另一个网络中的用户地址，由你到达该用户所经历的中间系统名组成。每一个中间地址按地址顺序写在用户系统的前面，相互用感叹号分开。假如你是在violet，而你想发信件到bell系统的chris，那么需要说明信件所经过的中间系统。对于中间系统为bell的地址将是stan!bell!chris。如果要发信件到rose系统的dylan，你要经历三个中间系统，就需要在地址中说明三个中间系统。在下面这个例子中，信件通过中间系统到达最终目的地。在第一个命令中，信件发到stan系统，然后发到bell系统，chris 在此系统上。在第二个命令中，信件首先被发到lilac，然后传到sf，再传到rose系统上的dylan：

```
$ Mailx stan!bell!chris < mydata
```

```
$ Mailx lilac!sf!rose!dylan < mydata
```

1.2 发送和接收邮件

要发送和接收电子邮件，你可以使用邮件程序，如：mailx 或 Elm。本章描述发送和接收信件的基本操作。采用mailx，是因为mailx已经成为了大多数Unix系统的标准工具。

要发送信件，键入mailx 命令，后跟接收信件用户的地址。在按回车后，出现提示，让你输入主题。可以键入一个信件描述(一个或多个词)，然后按回车，进入输入模式，在此键入信件内容，按回车键换新行，完成后，按Ctrl-d，即可以结束编辑，并发送信件。在键入Ctrl-d后你将看到屏幕显示字符EOT(结束符)。下面的例子中，用户发信件到地址为dylan 的用户，信件主题为Game。在键入信件文本后，用户按Ctrl-d。

```
$ mailx dylan
```

```
Subject: Game
```

```
Hockey is the best  
not tennis.
```

```
^D
```

```
EOT
```

```
$
```

发送给用户的信件被放入用户的邮箱里，一直到用户检索它们为止。若检索信件，可以使用mailx命令，此时不跟地址，仅输入mailx命令本身。它可以启动mailx界面，显示接收信件的头部摘要列表。信件头部摘要由一些字段组成，包括信件状态、信件序号、日期、大小和信件主题。信件的状态由一个大写字符如N或U标示。字符N表示一个新信件，字符U表示以前未读的信件。信件序号用于在mailx命令里代表信件，它在信件状态字段的后面。

```
$ mailx
Mailx version 5.5. Type ? for help.
"/usr/mail/justin": 3 messages 3 new
>N 1 Justin Mon May 11 11:31:19 5/44 "Car"
  N 2 Larisa Tue May 12 9:14:05 28/537 "Homework"
  N 3 chris Fri May 15 10:43:51 4/76 "Game"
?
```

在邮件提示符下，简单地键入一个信件序号，该序号所代表的信件将一屏一屏地显示，按空格键或回车键到下一屏。例如，在邮件提示符下，如果你键入数字3，就显示第一封信件。

```
Mailx version 5.5. Type ? for help.
"/usr/mail/dylan": 3 messages 3 new
>N 1 Justin Mon May 11 11:31:19 5/44 "Car"
  N 2 Larisa Tue May 12 9:14:05 28/537 "Homework"
  N 3 chris Fri May 15 10:43:51 4/76 "Game"

? 3
From chris Tue May 15 10:43:51 PST 1998
To: dylan
Subject: Game
Status: R
We have to figure out how to play hockey on a train
```

Chris

?

在邮件提示符下，键入q退出mailx。被保存的信件自动存入你起始（home）目录中一个名为mbox的文件中。可以用mailx加-F选项重新显示它们。

```
$ mailx-F mbox
```

1.3 收到邮件的通知

当信件被接收时，它们被放在一个文件里，该文件的操作就像邮箱一样，每一个用户由他自己的邮箱去保存等待的信件，通常情况下，一个信件到达后不会自动通知你。然而，可以采用from和biff实用工具，在有信件到达时通知你。

from程序列出你接收的、等待去读的信件。每一项显示发件人的地址和信件接收的时间。要使用from，键入命令from，并按回车键。

```
$ from
1 From justin Mon May 11 11:31:19 1998
  Subject: Car
2 From larisa Tue May 12 9:14:05 1998
  Subject: Homework
3 From chris Fri May 15 10:43:51 1998
  Subject: Game
$
```

当新信件接收到后，biff程序立即通知你。因此，当你希望一个信件到达后，马上得到通知的话，它是很有帮助的。使用biff命令带上y参数，即“biff y”，就可以打开biff。要关闭它，键入“biff n”。如果使用无任何参数的biff，就将显示biff是打开或关闭的信息。无论信件何时到达，biff都将显示一个信件通知，不管你此刻在做什么。如果你正在进行编辑任务，biff就中断编辑任务，在屏幕上显示信件通知。下一个例子，用户首先设置biff打开。然后，biff通知用户信件已被接收。接下来，用户检查是否biff仍然是打开的。

```

$ biff y
$
New mail for dylan has arrived:
-Date: Wed Mar 17 10:15:20
From: larisa
To: dylan
Subject: Vacation
    Dylan,
    Tell Justin his idea was brilliant???
...more...
$
$ biff
is y
$

```

如果你不想被中断, 可以使用“mesg n”命令阻止任何信件显示在屏幕上。“Mesg n”将不仅停止任何write和talk信件, 它同样也停止biff通知信件的到达。

1.4 自动接收邮件

如果你在休假, 在一定时间里就不能登录并阅读邮件。那么, 可以让发送到的邮件自动保存在你的邮箱里, 并回复发件人, 告知你不在。vacation命令用来完成这些操作, 读并保存你的邮件, 发送vacation信件给发件人。接收到的信件放置在你起始目录的.mailfile文件里, 发件人日志保存在.maillog文件里, 标准回答信件在系统/usr/lib/mail/std_vac_msg文件里。

vacation命令有几个选项, 如表1-2所示。-m选项用来指定你想保存信件的特殊邮箱, 可以简单地存取此文件, 列出所有你不在时所收到的信件。-M选项用于撰写你自己的休假信件, 缺省时只告知你在休假。-M选项可以用文本文件作为参数, 可以用编辑器建立一个文件, 组成你的信件。下面的例子, 名为vacplans的文件被用于在休假期间回答信件。任何接收到的信件被存入名为vacmsgs的文件里:

```
$ vacation -M vacplans -m vacmsgs
```

键入下列命令, 就可以使休假功能无效:

```
$ mail -F ""
```

表1-2 vacation选项

选 项	作 用
-d	增加日期到日志文件
-F 用户	如果邮件不能发到邮箱文件里; 就转发邮件到指定用户
-l 文件名	保存发件人文件名到指定文件(缺省为\$HOME/.maillog)
-m 邮箱文件名	保存接收的文件到指定文件(缺省为\$HOME/.mailfile)
-M 信件文件名	使用指定的信件文件作为假期自动回答文件(缺省为/usr/lib/mail/std_vac_msg)

1.5 二进制邮件和档案文件

通过电子邮件连接发送的信件可以看做文本信件, 信件由一系列字符组成。二进制文件, 例如被编译的程序, 就不能直接通过邮件程序发送。如果直接发送二进制文件, 它们到达后将不可用, 对于归档和压缩文件也是如此。