

第一部分 简介

第1章 UNIX操作系统

UNIX对工作站、微型计算机、大型机、甚至超级计算机等各种不同类型的计算机来说是一种标准的操作系统。虽然早在1970年，UNIX就已经开发出来，因为它具有适应性和可变性，所以，多年来UNIX不断发展，并随着不同的需求以及新的计算机环境的变化而变化。只通过简单的改变就可以使它实现新的功能或支持旧的功能。同时，UNIX并不是必须和某一种类型的计算机捆绑在一起，而是可以轻松地适应任何类型的硬件而不损失标准特性。多年来在不同的UNIX版本中所加入的功能并未影响底层的标准化。UNIX的众多版本之间只有微小的改变，并且工作模式极其类似。但是在UNIX的主流版本之间仍然有很大的区别。两个主流版本就是著名的系统V和BSD UNIX。系统V是由AT&T贝尔实验室开发，而BSD UNIX是由加州伯克利分校研制。这两个版本的UNIX经过集成成为现在的系统V第四版，简称SVR4。所以为了覆盖这两个UNIX版本，本书将集中介绍SVR4。虽然SVR4引入了两个UNIX版本中的绝大部分功能，但是本书也涉及了它们之间的区别。

作为介绍UNIX的一种方式，本章将把UNIX作为一个操作系统来讨论，包括UNIX的发展历史、UNIX系统设计以及各种UNIX版本，本章也将讨论如何最佳地使用本书。接触UNIX的人均有不同的背景知识，所以，有些功能可能会看起来似曾相识，而有些功能也许会使人感到陌生。本书将以一种有组织的上下文关系来介绍UNIX的不同特征，从而使您对UNIX有一个清晰的理解，不管您具有何种层次的知识背景，都将如此。

1.1 操作系统和UNIX

操作系统是一个为用户管理计算机硬件和软件的程序。起初，操作系统只是设计成能够执行重复操作的硬件任务。这些任务以管理文件、运行程序和从用户接收命令为中心。文件是存放于硬盘、软盘、磁带或压缩盘等存储设备上的电子数据。大多数用户需要计算机执行诸如从磁盘驱动器上读取和写入文件或在打印机上打印文件等硬件任务。如果没有操作系统，用户需要生成并运行一个程序以完成这些工作。然而，在一个操作系统中，用户则可以通过简单地向操作系统发送一条指令来执行这种任务。在操作系统中已经包含了像从磁盘读取文件这样的重复操作的设计。这种操作系统的基本功能如图1-1所示。

操作系统还具有能够接收并解释用户指令的接口。这个用户接口允许用户与操作系统交互。简单的操作系统的用户界面只允许输入命令行，复杂的则可以选择菜单和图标。

操作系统也管理应用软件。为了执行编辑文档或科学计算等任务你需要一些特殊的应用程序。编辑器就是一个应用程序的例子，用它可以编辑文档，改变或增加文本。而编辑器本身就

是一个包含指令的程序。要使用这个程序，应首先把它加载到计算机内存中，然后它的指令被执行。操作系统控制着所有程序的加载和执行，当然也包括任何的应用软件。在需要编辑器的时候，则可以简单地指示操作系统加载编辑器程序并执行。

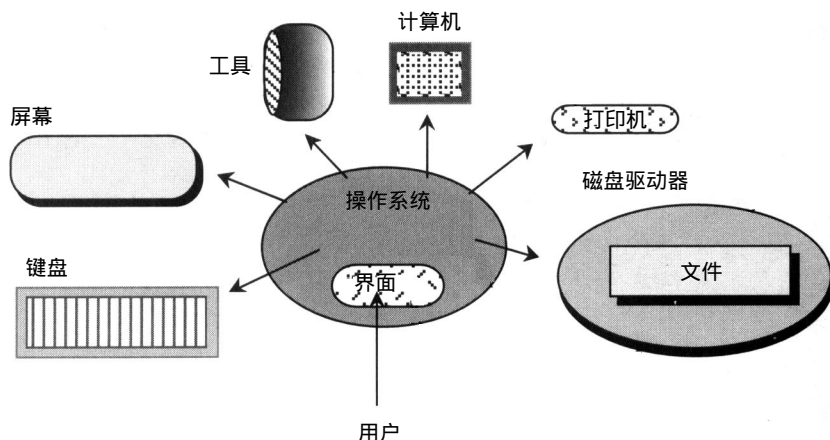


图1-1 操作系统及其管理的模块

文件管理、程序管理和用户界面是所有操作系统共有的传统特征。除此之外，UNIX又增加了另外两个特性。UNIX是一个多用户、多任务系统。作为一个多任务系统，用户可以请求系统同时执行多个任务。在运行一个作业的时候，可以同时运行其他作业。例如，在打印文件的同时可以编辑文件，而不必等待打印文件完毕再编辑文件。作为多用户系统，多个人可以同时使用该系统。几个用户可以同时登录到系统上，每人都可以通过自己的终端与系统交互。

操作系统起初的设计原则是使硬件能高效率工作。在计算机开发出来的时候，能力很有限，所以操作系统必须充分利用计算机的硬件能力。从这个角度上说，操作系统是从硬件而不是从用户的角度设计的，这迫使用户必须遵循硬件效率的要求。

另一方面，UNIX具有充分的灵活性，从根本上说它是一个科学研究系统。UNIX操作系统是由肯·汤普逊（Ken Thompson）在贝尔实验室于1969年开发成功的。起初，UNIX是为科研人员设计的操作系统。一个主要的目标就是生成一个系统以支持科研人员不断变化的需求。为了实现这一点，汤普逊将系统设计成能够处理很多不同种类的任务。所以灵活性则变得比硬件效率更为重要。虽然像UNIX这样灵活的系统并不一定比那些更加灵活的与硬件相捆绑的系统快，但是UNIX能够处理用户所能遇到的各种各样的任务。

这种灵活性使UNIX成为用户可用的操作系统，用户不是只限于和操作系统进行有限的、固定的交互，相反，操作系统可以为用户提供一套强大的工具，而且用户可以配置并对系统编程以满足他们的特殊需求。从这个意义上说，UNIX是一个面向用户的操作系统，它是一个操作环境。

1.2 UNIX的历史

在60年代后期，人们致力于开发新型操作系统技术，大部分工作集中在“时间共享”上。那

时候，大部分操作系统依赖与它们所运行的计算机硬件结构。一台计算机拥有一次运行一个程序的处理单元，操作系统可以将不同程序连接起来然后顺序地执行。这种结构拒绝对计算机的任何实时访问，程序员可以提交一个程序，然后操作系统将此程序与其他程序一起以批处理的方式一个一个地运行。只有当所有程序运行完之后，程序员才能得到结果。批处理系统极大地减慢了程序开发速度，而且改正和变动必须等到下一个批处理程序运行时方可进行。

分时系统允许程序员交互式地共享计算机的处理单元。处理单元在用户之间快速地切换，以至于用户都觉得是自己在独占系统。实现时间共享有很多种策略。策略之一是利用所谓的闲置时间，在交互式的情况下，在计算机必须执行的任务与任务之间经常存在时间滞后。例如，计算机的处理单元会以比用户从键盘输入字符快得多的速度运行，一次敲击能够被很快地处理，而计算机则须花很长的时间等待下一次击键。当计算机等待下一个要运行的任务的时候则处于闲置状态，分时则利用这个事实并使用闲置时间在每个用户的任务之间切换。这种在用户之间的切换则会使用户觉得他们用的是自己的个人机，而实际上所有用户在共享一台计算机。

在60年代UNIX出现之前，已经有几个分时系统。欧洲的剑桥多重访问系统、兼容分时系统（CTSS）、多路复用信息计算系统（MULTICS）都支持分时操作。MULTICS的开发是由来自通用电气、AT&T贝尔实验室以及麻省理工学院的科研人员发起的。这个系统除了时间共享以外，还在文件管理、程序交互和多任务等方面引入了很多新的概念。对多任务来说，计算机不仅能够适应不同的用户，而且某个用户还能同时运行多个任务。

1969年，在AT&T贝尔实验室，Ken Thompson、Dennis Ritchie和其他的研究人员开发了一个操作系统，它引入了MULTICS项目的很多特征，这个新的操作系统称为UNIX，它能够适合科研环境的需求。UNIX系统建立之初就是一个用户负担得起的、高效的、多用户、多任务的操作系统。

UNIX系统随着越来越多的科研人员的使用在贝尔实验室变得非常普及。1973年，Dennis Ritchie与Ken Thompson合作开始用C语言改写UNIX系统代码。此时，Dennis Ritchie已经将C语言开发成了一个灵活的程序开发工具，C语言的优势之一就是它能够通过一组通用的编程命令直接访问计算机的硬件结构。到这个时候，操作系统必须由一种计算机硬件相关的汇编语言改写。Dennis Ritchie和Ken Thompson使用C语言只编写一个版本的UNIX操作系统，这个版本可以在不同计算机上的编译器编译。实际上，UNIX操作系统变得可移植，所以能够在各种不同计算机上运行，而不需要或只需极少的程序改写。

UNIX逐步地由一种科研工具发展成为一种标准的软件产品，并由AT&T贝尔实验室发布。起初，UNIX只被当作一种科研产品，第一个UNIX版本是免费向很多知名大学的计算机科学系发布的，在70年代发布了几个版本的UNIX，直到1979年发布了版本7。

70年代中期，加州大学伯克利分校成为主要的UNIX开发者。伯克利在系统中加入了很多的功能，后来都成为了标准。1975年，伯克利通过它本身的发行机构，就是有名的BSD，发布了自己版本的UNIX。BSD UNIX后来成为了美国国防部高级研究工程局（DARPA）的研究项目的基础，BSD包括了强大的文件管理功能以及基于TCP/IP网络协议的网络特征。BSD4.2版已经广泛发布并由连同SUN公司在内的很多生产商采纳。SUN后来又开发了自己版本的UNIX，即SunOS。

BSD UNIX 成为了 AT&T 贝尔实验室版本的主要竞争对手，同时，其他的独立开发版本的 UNIX 也在萌芽。1980 年，Microsoft 和 SCO 发布了 PC 版的 UNIX，即 Xenix。1983 年，AT&T 发布了一个商业版本的 UNIX，系统 V 版本 1。

随着几个先进版本的 UNIX，如，系统 V、BSD 和 SunOS 等的发展，UNIX 成为一种关键的商业支持软件产品。AT&T 将 UNIX 转移到一个新的组织叫 UNIX 系统实验室，将来的系统 V 发行版本可以向上兼容，BSD UNIX 的很多特征被并入后来的系统 V。1991 年，UNIX 系统实验室发布了系统 V 版本 4 (SVR4)，它并入了 BSD、SunOS 和 Xenix 中的诸多特性。

系统 V 后来卖给了 Novell，再后来卖给了 SCO，SCO 进行着当前的维护和发行。很多其他的软件公司也开发了自己版本的 UNIX。如，IBM 的 AIX，Hewlett-Packard 的 HP-UX，Apple 的 A/UX，DEC 的 ULTRIX 和 DEC OSF/1，Silicon Graphics 的 IRIX，Sun 微系统公司的 Solaris，SCO 的 BSDI 和 SCO UNIX 是一些主流版本。最近，又出现了一个在 PC 和 Mac 等微型机系统上运行的免费发行版本 Linux，很多版本是基于 SVR4 同时并入了 BSD UNIX 的主要特征。BSDI 直接基于 BSD UNIX，目前可在微型机系统上使用。

所有这些版本的 UNIX 反应了 UNIX 能够很容易地适应不同应用的事实。UNIX 能够通过修改以集中在某种特殊任务上，这也反应出 UNIX 的起源是一个科学研究环境。在家中上 PC 和上超级计算机是等同的，UNIX 可以适用于管理商业操作或支持图像处理工作，同时，大部分的版本都包括同样的核心命令、特征和应用。

1.3 UNIX 标准

虽然目前有很多不同的 UNIX 版本可用，但开发商都在致力于一种通用标准。IBM、Hewlett-Packard、NeXt、Apple 和 Sun 分别支持不同版本的 UNIX，但它们都具有大部分的共同特征。甚至两种相互竞争的用户图形界面，Motif 和 Open-Look 也被集成为一种新的图形用户界面标准，叫作公用桌面环境 (CDE)。

IEEE 的第 1003 号项目，是为了开发一种独立的 UNIX ANSI 标准，这种新的 ANSI 标准 UNIX 叫做计算机环境的可移植操作系统界面 (POSIX)。这个标准定义了类 UNIX 系统如何操作以及系统调用和接口等的细节内容。

1.4 UNIX 简介

UNIX 通常可以分为四个主要部分：内核、shell、文件结构和应用程序。内核是运行程序和管理磁盘、打印机等硬件设备的核心程序。shell 则提供了用户接口，它从用户接受命令并将命令发给内核执行，文件结构则负责组织文件在磁盘等存储设备上的存储方式，文件是按目录的方式管理，每个目录可以包含任何数目的子目录，每个子目录可包含文件。

内核、shell 和文件结构共同构成了操作系统的基础结构。通过这三个模块，可以运行程序、管理文件以及同系统交互。另外，有一些外加的软件程序，即应用程序，也逐渐被认为是 UNIX 的标准特征，应用程序是一些特殊的程序，比如，编辑器、编译器和通信程序等，它们都执行标准的计算机操作。用户也可以生成他们自己的应用程序。

计算机、内核、shell 以及应用程序之间的关系可以描述成一系列同心圆，如图 1-2 所示。这

些同心圆说明了在用户和计算机之间的层次结构。在中间，是计算机本身，包括打印机、磁盘驱动器和其他的外围硬件设备。内核控制着硬件、程序的运行以及文件存储。shell与内核交互，把从用户接受的命令发送给内核。用户只需要与shell通信，而不会直接与内核通信，利用shell，用户可以运行不同的程序，如编辑器或通信程序。一系列标准程序就是所谓的应用程序。

1.4.1 内核

内核是控制计算机硬件的核心程序。它也实现了UNIX的多用户、多任务机制。内核将用户命令和程序组织为可以由计算机处理单元轮流工作的进程。虽然内核从某种程度上说是一种标准，但它也可以适应于特殊的需求和系统。内核是由C语言开发的，并且可能获得源码，可以通过修改源码并重新编译以获得新版本的UNIX。这种易于访问UNIX操作系统内部结构的特性是一种优势，同时又是一个弱点，它可以加速基于UNIX的研究和开发，同时它允许不同版本的简单增殖。由于这种原因，标准就逐渐成为一个问题，很多UNIX生产商已经经过多年的努力以制定UNIX标准。

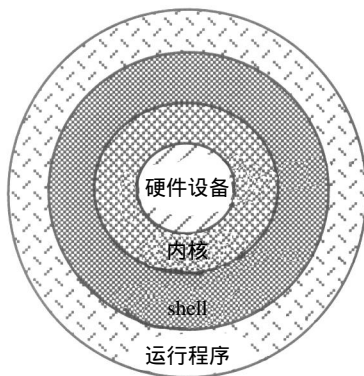


图1-2 UNIX操作系统结构

1.4.2 shell简介：Bourne、Korn、C-shell、BASH、TCSH和Z-shell

shell提供了一个内核与用户之间的界面。可以把它描述成一个解释器。它首先解释用户输入的命令然后发送给内核。可以把它想象成一个把命令翻译成内核操作格式的语言处理器。

shell界面非常简单。它通常包含提示用户输入命令的提示符，并且以回车键结束命令。就像用户在一个线路上输入命令，这条线路就是所谓的命令行，虽然由命令行输入的命令可能很复杂，但用户主要以命令的方式与系统交互。后来的UNIX版本为系统加入了图形用户界面（GUI）。Motif或Open-Look等的GUI使用图标和菜单来标识命令。GUI只是一个shell的前端，实际上是由shell解释命令并发送给内核的。图1-3则示出了shell、内核和其他系统模块之间的关系。

shell除了解释命令外，还有其他的工作。它也可以配置和编程。shell拥有自己的语言并允许用户编写程序并以一种复杂方式运行。shell编程语言具有许多常用编程语言的特征，例如，循环和控制结构等。用户可以生成像其他应用程序一样复杂的shell程序。

UNIX系统上每个用户都拥有自己的用户界面，即shell。用户可以使他们的shell界面适应于自己的需求。从这个意义上说，用户shell更像一个用户可以控制的操作环境。

多年以来，出现了很多不同类型的shell。主要有三种：Bourne、Korn和C-shell以及每一种的加强版：BASH、Z-shell和TCSH。Bourne shell是由贝尔实验室开发并成为系统V的shell。C-shell是由BSD UNIX开发。而Korn shell是一个Bourne shell的增强版。SVR4引入了这三种shell，并允许用户按自己的喜好选择。BASH shell是由Korn和C-shell的升级版，同时加入了高级命令

行编辑功能。Z-shell是Korn shell的增强版，TCSH shell是C-shell的增强版。很多当前版本的UNIX提供旧shell的同时也提供这些新的shell。例如，Linux系统就使用BASH shell作为用户缺省shell，而不是通常的Bourne shell。

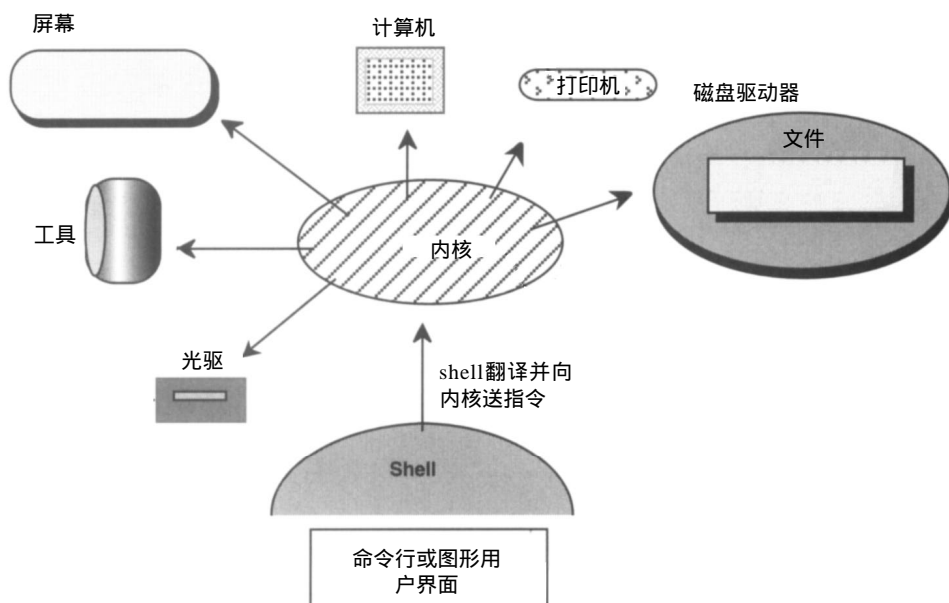


图1-3 shell和内核

虽然这些shell在命令级大部分兼容，但Bourne shell与C-shell在它们各自的编程语言上有着很大的区别。Bourne shell编程语言是一种更加通用的语言，这与C-shell有很大的不同，而C-shell编程语言非常类似于C编程语言，这也意味着它是一种复杂的语言。

1.4.3 文件结构：目录和文件

在UNIX中，一切都是文件，文件组织成目录。所以说，整个UNIX文件系统就是一个互相关联的目录集合，每个目录又包含文件。用户可以为自己的文件生成目录，并且可以把文件从一个目录移向另外一个目录。目录是一种组织和控制文件访问的很好的方式，用户可以为目录设置权限以允许其他人访问里面的文件或只允许自己访问。图1-4就是一个包含互相关联目录的文件结构的例子。

每个用户的目录最终都是与其他用户相连的。目录被组织成一种层次状的树型结构，以一个根目录开始，所有的其他目录最终都是从根目录起源的。图1-4是一个树型层次状文件结构的例子。用户可以遍历整个系统并进入任何一个打开的目录。这种文件结构的互相关联使得共享数据相当容易，从而使得几个用户可以访问相同的文件。

为了在文件上执行操作，用户可以使用拷贝、改名和删除文件以及列文件名和显示文件等命令，另外还可以使用目录管理命令生成或删除目录以及将文件从一个目录移向另一个目录，甚至可以将整个目录从一个目录移向另一个目录。

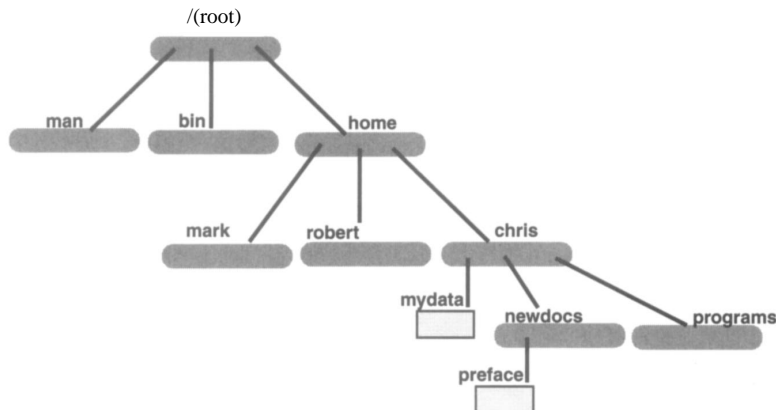


图1-4 目录和文件

文件都具有同样的物理实现，即字节流，就是一个连续的字节序列。没有性质上不同的文件，UNIX并不区分二进制和文本文件。文件中的信息可以当作二进制或文本处理，但文件本身拥有同样的标准格式。字节流格式同样适用于键盘和显示器等设备，当用户在键盘上敲入数据时，数据则被当作与文件一样的字节流标准格式来处理。类似地，从应用程序的输入输出拥有同样的字节流格式。

实际上，最好不要认为UNIX中的数据具有文件数据、键盘数据等不同的类型，而是只有一种类型，即字节流，并且这种类型的数据可以以多种不同的方式使用。它可以放入文件、从键盘输入、从应用程序输入输出。我们可以把数据想像为能够存为文件、输入应用程序或在屏幕上显示的流格式。因为数据是标准的，一个程序的数据可以容易的输入到另一个程序。数据流可以由一个程序移动到另一个正在修改的程序，并且最终存储在文件中。图 1-5则示出了同样的字节流数据能够用于不同的设备、程序或存储介质的例子。

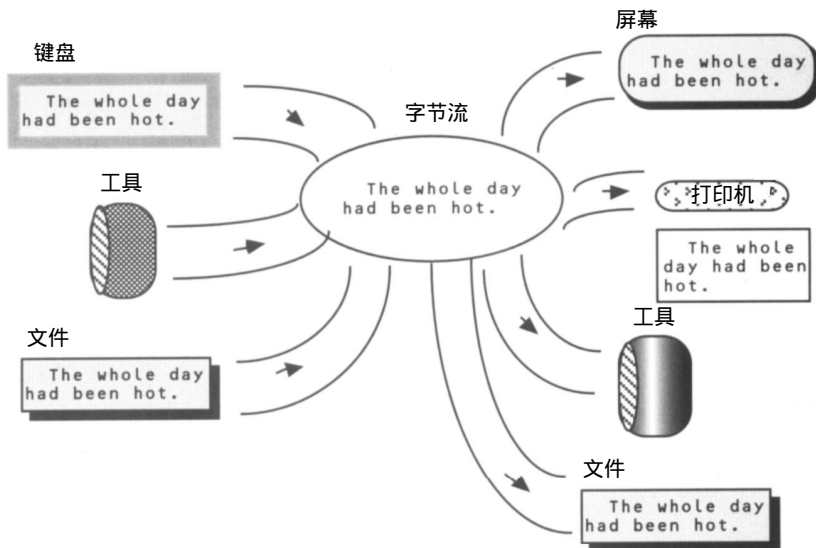


图1-5 字节流格式

1.4.4 应用程序：编辑器、过滤器和通信程序

UNIX包含大量的应用程序。其中一些是很简单的操作，另外一些则是非常复杂的程序并且拥有自己的命令集。很多的应用程序可以分为三类：编辑器、过滤器和通信程序，同时还有执行文件操作和程序管理的应用程序。

在所有的UNIX系统中有几种标准的编辑器：Ed、Ex、Vi和Emacs。Ed和Ex是行编辑器，而vi和Emacs是全屏幕编辑器。所有的标准编辑器都是为早期的、功能较弱的机器开发的，这些机器不能像现在的PC字处理器一样进行全屏幕移动光标操作。即使这样，Vi和Emacs拥有全屏幕功能，但它们并不能像很多鼠标驱动的字处理器那样流利简单的使用。Ed和Ex是一种每次只能编辑一行的行编辑器，即使有这种局限性，这些编辑器的功能还是非常强大的，它们拥有非常大的命令集并可以组合成很复杂的操作。

另外一组应用程序就是过滤器。过滤器从用户、文件或者其他数据源读取输入，然后检查并处理数据，最后输出结果。它们过滤通过它们的数据。有很多种不同类型的过滤器。一些是使用行编辑命令把已编辑的版本输出到文件。其他的则按一种模式搜索文件，然后将包括该模式的数据输出。另外一些则是进行字处理操作、探测文件中格式化的命令并将格式化的版本输出到文件。过滤器的输入不必非是一个文件，它也可以是用户的键盘输入。甚至可以是另外一个过滤器的输出，过滤器可以互相连接，一个过滤器的输出成为另一个的输入。用户也可以编写自己的过滤器。图1-6示出了由两个过滤器（cat、sort）处理的数据是怎样改变的。cat过滤器将两个文件的内容相连接然后把连接的数据发送给sort过滤器，sort对它进行排序，最后排序的数据被打印输出。过滤器是由所谓的管道相连接的，通过管道将一个过滤器的输出作为输入发送给另一个过滤器。

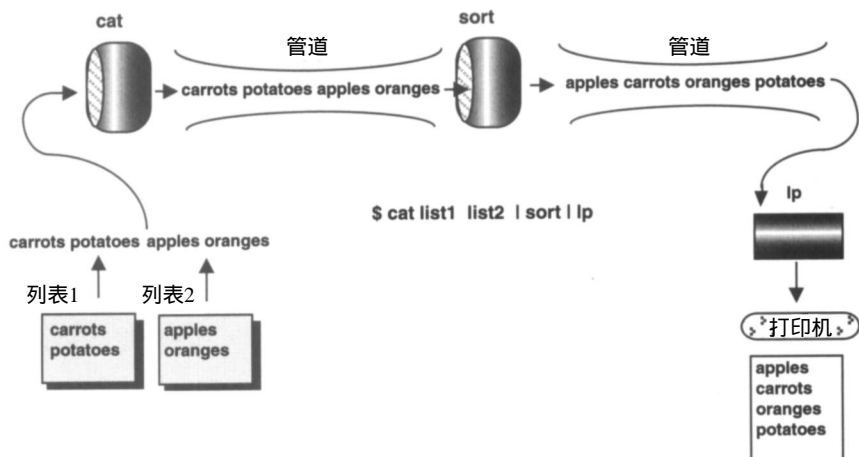


图1-6 过滤器

UNIX还拥有一套通信应用程序，通过这些程序可以与本系统或其他系统上的用户通信。作为一个多用户系统，UNIX必须与所有用户保持联系并对他们进行跟踪。基本的监视需求和基本的文件结构互连可以很容易地实现一个电子邮件系统，通过该系统用户可以从其他用户那里收

到或向他们发送消息，甚至可以同时向几个用户广播一个消息，或者与另外一个用户直接连接以建立一个实时会话。用户也可以通过网络与其他 UNIX系统上的用户通信，还可以远程登录到其他系统上并直接通过网络传输文件。

1.5 如何使用本书

本书旨在向读者详细介绍 UNIX但并不涉及过多的命令及编程。介绍 UNIX分三个阶段，首先是指南，然后介绍基本使用，最后是特殊应用。在第一部分，对 UNIX作了一个指导性的介绍（不需要任何计算机基础知识）。在第二、三部分，读者可以学到一些在文件管理、shell配置等日常工作中所需要的UNIX的基本功能。通过学习前三部分可以达到一个要成功使用 UNIX所必须具备的基本理解水平。每人都需要了解如何使用 shell和如何管理目录。第四、五部分包含了过滤器和编辑器的专题，从这两部分中，可以尽取所需。但为了在 UNIX上编辑文件必须至少了解一种编辑器。

第一部分是入门，在该部分中会讲解如何执行一些基本功能，如登录到系统、怎样使用图形用户界面（GUI）。还将介绍如何使用GUI登录与退出，以及如何用GUI管理文件、生成目录、拷贝文件和使用GUI编辑器，当然也可以用Vi访问文件。现在，很多人对于Windows和Macintosh这种图形用户界面比较熟悉，其实这些GUI与UNIX GUI之间有很多类似之处，然而，一定要记住UNIX GUI只不过是一个底层UNIX系统的一个界面而已，这一点很重要。

第二、三部分包含了每人必须了解的日常操作知识。这两部分提供了对命令行界面的详细讲解以及对文件结构的深入阐述。其中包括如何配置UNIX以满足自己的需求。UNIX是一个多用户系统，每个用户拥有自己的系统界面，并且可以按自己的需要进行配置，这个界面称为shell，用户可以使用特殊的shell命令和变量配置它。

第7章对Vi做了介绍。在很多系统中，Vi是标准编辑器，有时候也许不得不用它。然而，用户通常觉得它很难学。它是一个拥有很多复杂命令的、功能强大的编辑器。对于Vi我们只介绍一些常用的基本命令并对每个命令做了详细的解释。如果要了解其他的标准或非标准的编辑器，可以参考第5章中关于编辑器的介绍。这些都将放在本书的最后，因为编辑器越来越复杂，而一开始就对编辑器进行深入的分析恐怕会喧宾夺主。

第四、五部分则主要介绍了特殊的过滤器和编辑器。第四部分从介绍sed、diff等编辑型过滤器入手着重分析了几种高级过滤器。接着介绍了正则表达式并讨论了sort和uniq等数据过滤器。第五部分介绍了UNIX上不同的编辑器，如Vi、Ex和Emacs等。

在本书中，对于每一种特性，都提供常用命令的图表，在附录B中包括了所有这些命令的完整列表，当然也包括对每个命令选项的详细解释。

1.6 小结

UNIX是一个交互式的多用户、多任务的操作系统。UNIX起源于一个面向研究的分时系统，后来成为一个标准的操作系统，可用于网络、大型机和工作站。目前存在几种版本的UNIX，最通用的就是系统V和BSD。系统V版本4形成了一个集成的标准，它几乎并入了所有主流UNIX版本的特征。

UNIX系统是由内核、shell、文件结构和应用程序组成。内核是系统心脏。它包含了直接管理计算机硬件的控制程序。而 shell则是用户界面。shell从用户那里接收命令并发送给内核执行。shell能够适应于单个用户的需求，shell甚至拥有能够对命令进行编程的编程语言。

文件结构包含目录，目录里面又包含文件。目录提供给用户一个方便的方式来组织他们的文件，用户可以将一个文件或目录从一个目录移到另一个目录、设置目录的权限以及打开文件或与其他用户共享文件。

UNIX系统中存在一组标准程序，通常称为应用程序。应用程序一般可分为：编辑器、过滤器和通信程序。编辑器虽然功能强大，但使用起来可能要比目前很多的 PC编辑器困难。过滤器是一种接收数据并将该数据以另外一种形式输出的特殊应用程序。通信程序允许用户从其他用户接收信息或向他们发送信息，也可以通过网络向其他 UNIX系统上的用户发送信息。