

新编 UNIX 实用大全

莫文 刘看望 合编
毛兴平 程军



陕西电子杂志社

新编 UNIX 实用大全

莫文 郭春侠 刘看望

王兴平 程军

陕西电子杂志社

内容提要

UNIX 自问世以来, 经过 20 年的演变和发展, 其功能不断增强, 应用面越来越广。随着计算机的日益普及, 更多的计算机专业和非计算机专业人员正在对 UNIX 产生兴趣, 他们注视着 UNIX 的发展, 希望了解 UNIX 特点, 功能和用法。这正是我们编写此书的目的。我们希望读者通过此书的阅读, 能够掌握 UNIX 系统的基本功能和用法, 并能够进行较高级的应用开发工作。

本书首先回顾了 UNIX 的发展, 然后介绍了 UNIX 的交互环境, 多用户概念, 用户注册, 及 UNIX 文件系统等内容。Shell 是用户与 UNIX 核心间的接口, 书中用较大篇幅介绍了 Shell 环境及 Shell 编程, 书中还介绍了一些 UNIX 实用工具, 如 UNIX 编辑器, UNIX 邮件系统, 及 UUCP 等。为了使用户更好利用 UNIX 的资源, 本书也介绍了 UNIX 安全及系统管理方面的内容。最后介绍了 UNIX 下的 C 语言开发环境, 及其开发工具 (如 make, SCCS, lex 和 YACC 等), 以便为 UNIX 的程序设计提供支持。UNIX 的内容极为庞杂, 我们力图为您提供各方面的知识和技术, 希望您能满意。

本书第一章、第二章、第四章至第八章由刘看望编, 第三章由程军编, 第九章、第十章由王兴平编, 第十一章至第十八章由莫文、郭春侠编。

在本书的录入和校对过程中, 陈瑞英同志给予了极大的帮助, 深表感谢。

鉴于作者的水平有限, 书中难免有不尽完善之处, 恳请各位同行指正。

目 录

(20)		书文通普	0.2
(60)		书文叙特	0.2
(62)		书文特叙	0.2
(61)		书文思思	0.2
第一章 UNIX 简介			
1.1	UNIX 原型.....		(1)
1.2	UNIX 标准.....		(2)
1.3	UNIX 特点.....		(3)
第二章 UNIX 交互环境			
2.1	哑终端.....		(4)
2.2	智能终端.....		(4)
2.3	标准终端系统.....		(5)
2.4	终端类型的识别.....		(5)
2.5	UNIX 与终端.....		(5)
第三章 UNIX-多用户系统			
3.1	共享资源.....		(6)
3.2	网络.....		(6)
3.3	网络类型.....		(6)
3.4	规模因素.....		(7)
3.5	网络拓扑.....		(7)
3.6	网络共享和存取方法.....		(10)
3.7	UNIX 网络.....		(15)
3.8	IPC 和 UNIX 4.2 BSD 套接口.....		(41)
3.9	思考题.....		(54)
第四章 UNIX 的注册			
4.1	简介.....		(55)
4.2	注册.....		(55)
4.3	典型注册信息.....		(57)
4.4	UNIX 提示.....		(58)
4.5	用户目录.....		(58)
第五章 UNIX 文件系统			
5.1	简介.....		(60)
5.2	UNIX 文件系统的特点.....		(60)
5.3	UNIX 文件系统的概貌.....		(60)
5.4	目录文件.....		(61)
5.5	链接文件和 i-节点.....		(64)

目 录

5.6 普通文件	(65)
5.7 特殊文件	(66)
5.8 文件的特性	(66)
5.9 思考题	(73)

第六章 UNIX 物理文件结构

6.1 简介	(74)
6.2 概述	(74)
6.3 磁盘分区	(74)
6.4 磁盘块	(75)
6.5 i-节点结构	(76)
6.6 思考题	(78)

第七章 UNIX Shell 和实用程序

7.1 简介	(80)
7.2 交互式 shell	(81)
7.3 实用程序	(83)
7.4 输入和输出重定向	(110)
7.5 管道和过滤器	(117)
7.6 复杂模式和引号	(134)
7.7 命令综述	(139)
7.8 思考题	(143)

第八章 shell 元字符和变量

8.1 简介	(145)
8.2 shell 元字符	(145)
8.3 shell 变量	(155)
8.4 创建 shell 变量	(166)
8.5 命令综述	(172)
8.6 思考题	(173)

第九章 UNIX 编辑器

9.1 引论	(175)
9.2 ex 编辑器的使用	(175)
9.3 ed	(186)
9.4 Vi	(187)
9.5 思考题	(201)

第十章 UNIX 邮件系统

10.1	引论	(202)
10.2	Mail	(202)
10.3	用户别名	(208)
10.4	浏览邮件标题	(209)
10.5	选择消息	(210)
10.6	删除和打印消息	(211)
10.7	邮件通知-biff 命令	(211)
10.8	保存邮件-s 命令	(212)
10.9	转发邮件-f 命令	(213)
10.10	传递邮件-mb 命令	(214)
10.11	存消息正文-w 命令	(215)
10.12	打印消息	(215)
10.13	恢复消息-u 命令	(216)
10.14	退出制作模式	(216)
10.15	~ 命令	(217)
10.16	Mail 选项	(222)
10.17	串选项	(225)
10.18	Mail 命令行选项	(226)
10.19	mail 工具	(227)
10.20	邮件加密	(230)
10.21	直接通讯	(232)
10.22	思考题	(236)

第十一章 系统管理

11.1	简介	(238)
11.2	超级用户	(238)
11.3	拒绝用户访问	(259)
11.4	恢复一个被禁止的用户	(261)
11.5	生成新口令	(261)
11.6	维护充足的存贮空间	(263)
11.7	/etc/fstab 文件的结构	(268)
11.8	建立文件系统-mkfs	(271)
11.9	系统配置	(275)
11.10	常驻后台进程-daemons	(278)
11.11	UNIX 系统的文件备份	(280)
11.12	检查文件系统-fsck	(284)
11.13	preen	(288)

11.14	重启动 UNIX 系统-reboot	(289)
11.15	关机-shutdown	(289)
(505)	11.16 关机过程的顺序	(290)
(505)	11.17 启动 UNIX 系统	(290)
(805)	11.18 与用户保持联系-/etc/motd	(291)
(905)	11.19 思考题	(291)

第十二章 UNIX 安全

(115)	12.1 简介	(292)
(515)	12.2 UNIX 在安全性方面的存在问题	(292)
(815)	12.3 文件的安全性-chmod	(292)
(415)	12.4 SUID 设置	(294)
(215)	12.5 查找 SUID 文件-Find	(296)
(615)	12.6 伪登录	(297)
(015)	12.7 监视和记帐功能	(299)
(015)	12.8 可安装文件系统的安全	(302)
(715)	12.9 系统管理文件的安全	(302)
(555)	12.10 系统登录文件	(303)
(255)	12.11 思考题	(303)

第十三章 shell 编程

(055)	13.1 简介	(305)
(555)	13.2 条件命令-if	(305)
(055)	13.3 测试命令-test	(306)
	13.4 Case 和 Switch	(311)
	13.5 循环-for 和 foreach	(313)
(855)	13.6 循环-while 和 until	(316)
(855)	13.7 退出和继续循环的命令	(318)
(955)	13.8 其它循环命令-xargs	(319)
(105)	13.9 捕获中断-trap	(320)
(105)	13.10 更新数据	(321)
(805)	13.11 AWK	(324)
(805)	13.12 思考题	(330)

第十四章 C 语言程序开发环境

(875)	14.1 C 标准库	(331)
(085)	14.2 系统调用	(333)
(485)	14.3 C 语言编译器-CC	(333)
(885)	14.4 C 程序的检查-lint	(338)

(104)	14.5 C 程序的检查程序-lint	(345)
(104)	14.6 调试工具-adb 和 sdb	(346)

第十五章 软件构造和维护工具-make 和 SCCS

(114)	15.1 软件自动构造与维护工具-make	(348)
	15.2 源代码控制系统-SCCS	(354)

第十六章 词法和语法分析程序

	16.1 词法分析程序的生成程序 -lex	(360)
	16.2 编译程序的编译程序-YACC	(367)

第十七章 进程通信机制

	17.1 system 系统调用	(376)
	17.2 Exit 系统调用	(376)
	17.3 Exec 系统调用	(377)
	17.4 Exec 命令与元字符	(378)
	17.5 fork 和 wait 调用	(378)
	17.6 信号	(379)
	17.7 进程挂起	(381)
	17.8 超时检测	(381)
	17.9 非局部转移	(383)
	17.10 无名管道	(384)
	17.11 管道函数	(384)
	17.12 使用低级管道	(385)
	17.13 有名管道	(388)
	17.14 报告 IPC 功能状态	(389)
	17.15 删除 IPC 功能的信息	(389)
	17.16 消息队列	(389)
	17.17 共享存储区(SM)	(394)
	17.18 信号灯	(395)

第十八章 基本通信软件 -UUCP

	18.1 UUCP 的字装	(404)
	18.2 连接类型	(404)
	18.3 UUCP 命令	(404)
	18.4 UUCP 的文件	(405)
	18.5 文件 passwd 和 inittab	(407)
	18.6 特殊文件的访问模式	(407)
	18.7 实例	(407)

(343)	18.8 文件传送	(409)
(346)	18.9 错误及调试	(409)
	18.10 删除所接收信息	(410)
	18.11 使用 CU 命令拨叫另一个系统	(411)

(348)		12.1 软件自动检测工具-make
(344)		12.2 源代码编译系统-SCCS

第十六章 网络分析工具

(360)		16.1 网络分析工具的主程序-lex
(363)		16.2 网络分析工具的主程序-YACC

第十七章 进程通信

(376)		17.1 system 系统调用
(376)		17.2 Exil 系统调用
(377)		17.3 Exec 系统调用
(378)		17.4 Exec 命令与子程序
(378)		17.5 fork 和 wait 调用
(379)		17.6 信号
(381)		17.7 进程挂起
(381)		17.8 超时检测
(383)		17.9 非阻塞操作
(384)		17.10 无名管道
(384)		17.11 管道函数
(385)		17.12 使用低级管道
(388)		17.13 有名管道
(389)		17.14 报告 IPC 功能状态
(389)		17.15 删除 IPC 功能的信息
(389)		17.16 消息队列
(394)		17.17 共享库 (S.M)
(392)		17.18 信号机

第十八章 基本通信软件-UUCP

(404)		18.1 UUCP 的术语
(404)		18.2 连接类型
(404)		18.3 UUCP 命令
(405)		18.4 UUCP 的文件
(407)		18.5 文件 passwd 和 initrd
(407)		18.6 特殊文件的访问模式
(407)		18.7 实例

第一章 UNIX 简介

UNIX 自从诞生迄今已有二十多年，其发展史很值得我们回顾，在这里我们仅作一简单的介绍。UNIX 是一个大型而复杂的系统，但我们不能由于这种理由而拒绝它。如果你对 UNIX 的诞生比较感兴趣，那么就从本章开始学习，否则，可以从第二章开始学起。

1.1 UNIX 原型

UNIX 是一个计算机操作系统的名字，是贝尔实验室所拥有的注册商标。UNIX 不单是一个系统，它也是一类相关系统的通用名字。

在六十年代后期，贝尔实验室致力于一个交互式多用户系统 Multics 的开发。1968 年，由于该系统新出现的一些问题，贝尔实验室的计算机科学与研究部从该工程中退出。

从 1969 年开始，曾工作在 Multics 工程的职员决定寻求一种新的系统。1969 年，k.To-mpson，D.Ritchie 和 R.Canaday 为一新的操作系统设计了一文件结构，它便成为今天 UNIX 系统的雏形。也就在 1969 年，Thompson 开发了一个 'Space travel'（太空旅行）的空间仿真程序，但该系统因机器本身的限制而难于控制，它最初的设计基于 GE-645 分时系统。

此后，Thompson 开始使用 PDP-7 系列计算机，他和 Ritchie 重新编写了运行于新的计算机上的 'Space travel'。Thompson 也开始在 PDP-7 上实现过去 Multics 的文件系统，并且开发了一系列文件处理所必须的小型实用程序，如文件拷贝等。

这就是 UNIX 系统的第一次移植：从一个已存在的系统移植到一个新计算机系统上。在 1970 年，Brian kernighan 建议用 UNIX 作为该系统的名字。UNIX 这一词语的出现在最初引起极大的争论，许多人相信它是早期 Multics 系统的命名。

此时 kernighan 和 Ritchie 已开发了 C 语言，它是一种比较灵活的语言。新的 UNIX 系统除了核心用汇编语言编写外，几乎所有的代码都用 C 语言来实现。也正是由于这种原因，UNIX 的实现独立于任何机器，它具有非常好的移植性。这也是 UNIX 在世界范围内取得重大成功的最主要的原因。

在 70 年代最初的几年中，UNIX 仍然是贝尔实验室的私有财产，直到 70 年代中期，UNIX 才向学术界公开发表。Thompson 作为伯克利加利福尼亚大学的名誉教授，在计算机科学系开始介绍 UNIX 系统。贝尔实验室也支持以这种方式促进 UNIX 系统，并同意在美国和欧洲宣传推销 UNIX 系统。

UNIX 系统的发展至少有三个主要的流派。图 1.1 给出了变革的概貌，特别是导向 4.3BSD 和导向系统 V 的那些分支。

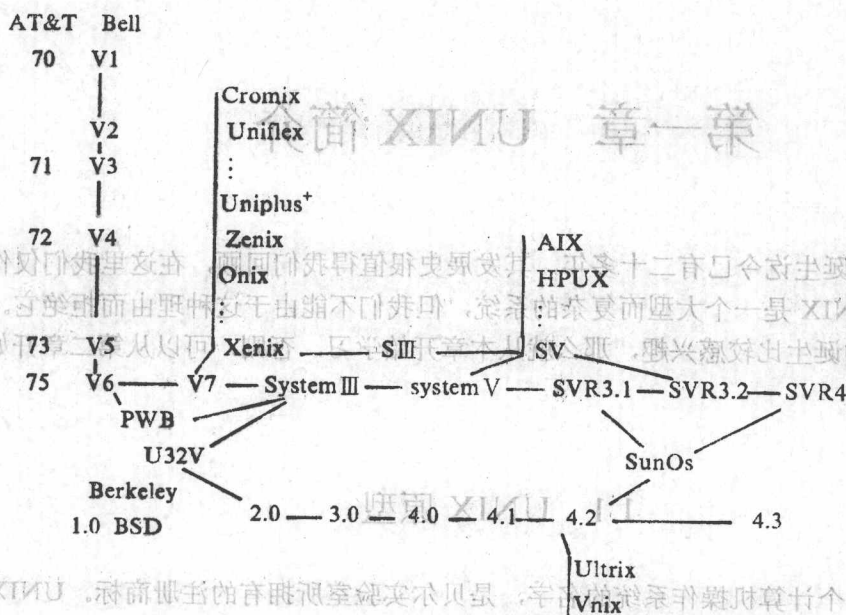


图 1.1 UNIX 演变示意图

把 UNIX 用作一个程序开发环境，有两个主要因素使它得以在商界流行。首先，是 UNIX 在大学的推广。到 1980 年，UNIX 已在美国和欧州的许多大学和学院里讲授，并且支持着许多计算机专业的课程，对操作系统这门课程尤为突出。

那些在学校受到过 UNIX 培训刚进入工作岗位的新毕业的大学生，支持把 UNIX 作为一个计算机系统。由于 UNIX 的优点，而且由于这些毕业生在校期间学习过 UNIX，因而，他们对它很有信心，并且自作主张将 UNIX 安装在他们那里，并把他们的工作方法带进老的数据处理领域。

另一个使 UNIX 进入商业领域的事实是 AT&T 公司的解体，使得 Bell 实验室从中分离出来。Bell 实验室不再受 AT&T 公司的约束，这样也推动了 UNIX 进入商业领域。

1.2 UNIX 标准

那么，一个 UNIX 版本成为一个真正的工业标准的可能性是什么呢？不像其它的操作系统，如 MS-DOS，它由一个单个的系统供应商控制着，而 UNIX 则散布在硬件供应商，UNIX 软件公司以及 AT&T 公司自身中。UNIX System V 已被认为是其它 UNIX 版本必须遵循的基本标准。如今，许多商品化 UNIX 系统都是基于 System V 的。

因为存在这么多不同类型的 UNIX 和类 UNIX 系统，因而妨碍了 UNIX 标准版本的形成。这是由于各竞争者为了占领这一巨大的市场而不断开发 UNIX 的增强版本而引起的。

system v 的版本 4.0，最近已经发行了。它兼备三种较流行的 UNIX 版本—microsoft

公司的 UNIX 版本 xenix, 它流行于个人计算机上; 旧的基本系统 system V 和 Berkecey UNIX。所有这三种 UNIX 系统在编写本书时都已使用。

第二章 XINU

1.3 UNIX 系统的特点

同其它操作系统相比, UNIX 操作系统有三大特点:

(1) 系统用高级语言书写, 移植性好。

UNIX 系统的大多数源码是用 C 语言编写的。以前, 一般均认为操作系统必须用汇编语言开发, 以便使硬件访问取得最高效率。由于 C 语言本身是一种恰当好处的高级语言, 它既是一种高级语言, 可以很容易地在广泛类型的计算机硬件上编译; 又不是很复杂或限制很大, 使程序员必须转去用汇编语言编写以取得合理的效率或功能。UNIX 操作系统的百分之三 (需访问硬件的部分, 如环境切换等) 需要用汇编语言编写。虽然 UNIX 的成功并不单是由高级语言编写而带来的, 但是用 C 语言是关键的第一步。C 语言是继承了 B 语言而导出的, 而 B 语言是从 BCPL 发展而来的。近年来, C 语言又有了一个变种 C++, 它是一种面向对象的高级语言, 能很容易实现面向对象的程序设计。

(2) UNIX 早期版本以源码形式发布。由于提供了源码, 系统创立者确保了其它机构可以不仅使用 UNIX 系统, 而且可以深入系统内部设法修改和补充系统。容易在系统中采纳新的思想是修改系统的关键因素。每当出现一个新系统企图超过 UNIX 时, 就会有人对新系统进行分析而把它的中心思想加入 UNIX。这种在充满新思想的环境中使用一个小型的, 用高级语言编写的, 容易理解的系统的独特能力, 使得 UNIX 系统不断发展, 并日趋完善。

(3) 并发进程及管道线

UNIX 使多个用户可以并发地运行多个进程, 并提供了多个进程连接为若干命令的管道线。原先只有在大型的昂贵的机器上运行的操作系统才有运行多个进程的功能, 并且并发进程的数目通常是由系统管理员严格控制的。

参考文献

1. 操作系统原理, 清华大学出版社, 1980。
2. UNIX 操作系统, 清华大学出版社, 1980。
3. 操作系统原理, 清华大学出版社, 1980。
4. 操作系统原理, 清华大学出版社, 1980。
5. 操作系统原理, 清华大学出版社, 1980。
6. 操作系统原理, 清华大学出版社, 1980。
7. 操作系统原理, 清华大学出版社, 1980。
8. 操作系统原理, 清华大学出版社, 1980。
9. 操作系统原理, 清华大学出版社, 1980。
10. 操作系统原理, 清华大学出版社, 1980。

第二章 UNIX 终端

为了与 UNIX“交谈”，我们需要介绍一些基本的知识。如果你能熟练的使用计算机终端，那么你可以跳过本章。

与 UNIX 通讯是通过终端来完成的。终端一般由视频显示器(VDU)和键盘组成。键盘是同 UNIX 系统通讯时的最主要的输入设备。

UNIX 是一个交互式系统。其含意是指，你通过键盘输入 UNIX 指令，然后它被发送到系统去处理。处理完成后，系统把处理结果再返回给你。

由于 UNIX 系统是交互式的，因而它也被认为是一个全双工系统。键盘使你可以输入命令，以便让 UNIX 处理它们。键盘上几乎所有的键都是可印刷字符（单个的数字或字符）。然而，不同于一般的可印刷字符，还有一些称作控制字符的字符。这些都是非印刷字符，像键盘上的 DEL 键以及光标移动键所产生的字符就是非印刷字符。在交互方式下，由于命令要在一行内输入，所以不能正常使用所有的光标键。在一些系统中，左、右光标键（←和→键）可用来修改命令行的内容。在我们的叙述中，命令行是指包含有当前要执行的命令的那一行。但使用屏幕编辑的情况例外。

在键盘上，RETURN(ENTER)键比较重要。在每条 UNIX 命令输入完毕后必须按一下此键。

2.1 哑终端

终端通常分为两类。第一类被称为哑终端。哑终端是无任何处理能力的简单终端，一般仅用作同主机通讯的双向通讯设备。

2.2 智能终端

终端的另一种类型是智能终端。它们具有附加的处理能力。智能型终端的一个重要的功能是，它能从 UNIX 系统获取信息，并可将这些信息存贮在与它们相连接的存贮设备上，如磁盘设备。

智能型终端能减轻主计算机负荷。大多数个人计算机可以通过终端仿真技术用作终端。以这种方式使用的个人计算机常被称为“灵巧终端”(Smart terminal)。

2.3 标准终端系统

计算机终端在字符显示和如何解释控制字符方面和个人计算机存在差异。幸运的是已经出现了一些标准的终端仿真器。VT100 以及其后出现的 VT200 / 300 系列终端已成为工业标准。

上面提到的 VT 系列终端最初由 DEC 公司制造。不过现在 VT100 型终端已用在许多别的计算机系统及终端仿真系统中。基本的 VT100 型终端可以看成是哑终端型的产品,这是由于它缺乏附加的处理能力。然而 DEC 公司也已生产出了智能型终端。如 VT200 是包含一定智能的终端,而 Rainbow 100 则是另一类较好的智能型终端。DEC 公司的大多数智能终端都有本地存贮和附加的处理能力,使得它们可以在本地运行其它计算机软件。

2.4 终端类型的识别

怎样判断一个终端是哑终端还是智能型终端呢?根据定义,智能型终端具有辅助的处理能力。这种辅助的处理能力可能表现为内部存贮设备。如果属于这种情形,终端必须将该设备装入其壳子中,这样将使终端体积加大。

然而有些终端的辅助处理能力是由附加的电路来完成的。这些电路设计得非常紧凑。如果你使用的终端是一台个人计算机,如 IBM AT 或 Clone 计算机,那么,它具有很强的处理能力,如存贮机制,内存电路等。这些一般都封装在系统单元中。

2.5 UNIX 与终端

终端是与 UNIX 相连的设备之一。UNIX 通过一具体的名字来标识它所使用的终端,名字是由字符串“tty”与一数字或字符组合而成。字符串“tty”找表“teletype”,它是早期“terminal”的同义词。

终端标识符中的数字部分有时代表着 UNIX 系统可支持的终端个数。例如,如果 UNIX 安装了 20 个用户终端,那么任何在 tty0~tty19 范围的终端标识符都将是合理的(注意:0 在许多 UNIX 配置中是有效的)。

终端的编码可能由于制造商的不同而不同。你可能发现有些 UNIX 系统用纯字符标识终端,像 ttya, ttyb 等。

另一个在许多 UNIX 中使用的终端名是“Console”,它表示该终端是供系统管理员使用的,它常被用来监视当前 UNIX 系统的每个终端。它也是用来维护天天运行的 UNIX 系统的终端。

明白 UNIX 可以支持非常广泛的各种各样的终端是非常重要的。在以后的章节中,我们将详细地讨论计算机终端和 UNIX 系统这一话题。

第三章 UNIX-多用户系统

UNIX 是一个多用户系统，其含义是指在任何时候，至少有多于一个的用户可以访问主机系统的资源。一个典型的 UNIX 系统，可能允许一百个或更多的用户同时登录。在任何时刻，所有这些用户能同 UNIX 通讯的可能性取决于安装的终端数量的多少，以及主机系统的配置情况。大多数 UNIX 系统的注册用户都多于终端的数量。

3.1 共享资源

由于 UNIX 是一个多用户系统，因而可以实现资源共享。UNIX 通过向系统中的每个用户分配时间段的方法来实现资源共享。时间段有时也叫时间片。当许多用户同时存取或处理数据时，UNIX 将在每个用户间快速的切换，以致在大多数情况下，用户根本感觉不到这种变化。

当 UNIX 处理一个任务时，其主要目标是试图完成它。而这依赖于该任务需要多少个处理进程以及该任务所分配到的优先级。如果任务较长而且比较复杂，UNIX 将转去处理另一个也需要关心的任务，并在该任务完成后重新返回原任务。这种切换处理也就是所谓的多任务，而 UNIX 系统管理所涉及到的一个主要部分也就是处理每个终端发出的各种各样的任务。资源共享是任何多用户系统的必不可少的组成部分，它由时间片分配和多任务机制来实现。

3.2 X 网络

所有的多用户系统都是一个网络系统。网络的优点是非常巨大的。例如：数据由于存储在中心海量存储设备上而不会重复，而打印机、磁盘机等外设可以被所有用户访问。

3.3 网络类型

网络的分类主要依据它们的形状和规模。一个网络的形状，或者用户终端互连的方法，属于网络拓扑（后面讨论）范畴。有两个主要的因素决定网络的类型，即形状和规模。

3.4 规模因素

网络的规模取决于用户的多少及其在地理位置上的分布。本节我们讨论两个主要的网络，它们取决于它们在物理位置的规模。

3.4.1 局域网(LANs)和广域网(WANs)

网络被分成两个主要的类，它们在很大程度上依赖于它们的地理范围的大小。这里我们考虑的两种网络是局域网(LANs)和广域网(WANs)。局域网是分布在一个较小地理范围内相连接的设备组（例如：个人计算机，工作站等设备）。局域网的数据传输率可以达到10M 位/秒。（更高的数据传输率可以通过光纤技术来获得，如 FDDI—Fibre Distributed Interface）。

广域网(WAN)是在一个较大地理区域内设备的互连。像公用电话交换网(PSTN—Public Switch Telephone Network)便是一个广域网的很好例子。广域网可以在方圆1000公里的范围内进行通讯。许多已存在的广域网的数据传输率已达100千位/秒。较新的数字网络技术，像集成服务数字网络(ISDN—Integrated Services Digital Network)允许采用更高的数据传输率（一般在144~2048千位/秒范围内）。

3.5 网络拓扑

下面我们将讨论四类主要的网络拓扑，以及它们各自的优缺点。

3.5.1 星型网络拓扑(star)

星型网络中，各终端间传输数据的所有工作都由网络的中心点来完成。所以这种系统的智慧都存在于其中心节点内。外围节点不必要求是较高档次的智能系统，这使得系统价格低廉。星型网络的主要缺点是，如果中心节点损坏，那么整个系统将停止工作。星型网络拓扑示于图3.1中。

3.5.2 环型网络拓扑(ring)

环型网络拓扑使智能型节点（或终端）散布在网络中，这样中心控制便不再需要。大多数环型网络都有一个监控站（一般是网络本身的一个节点）。它负责数据传输中的错误处理。这种类型网络上的数据传输一般是单向的，但有些环型网络系统可以配置成双向传送数据。环型网络拓扑示于图3.2中。

3.5.3 总线网络拓扑(BUS)

总线网络由总线和总线两端的终结器组成。（图3.3），因而它很容易识别。网络中的设备（例如一个终端）沿着总线发送数据和一个设备的地址。所有设备将快速的检查数据

并判断地址是否与其本身的地址匹配。这种类型的网络是“被动式”网络，其含义是指网络中的设备要侦听发送到它们自己的数据。

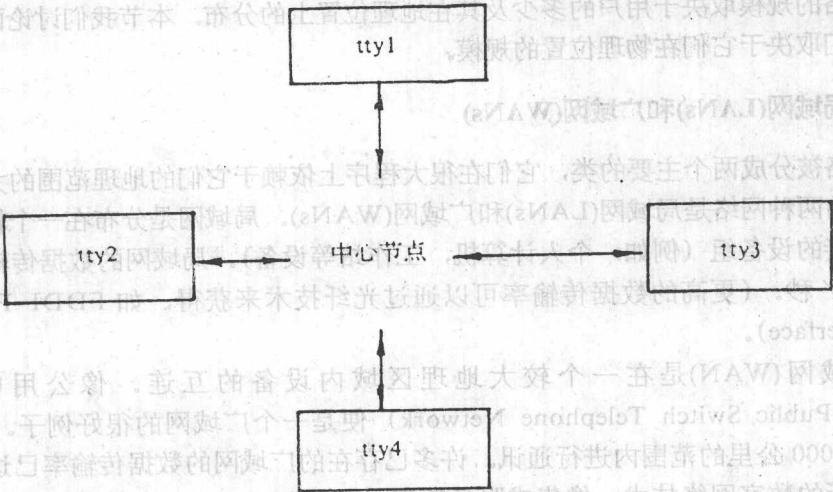


图 3.1 四个节点的星型网络拓扑

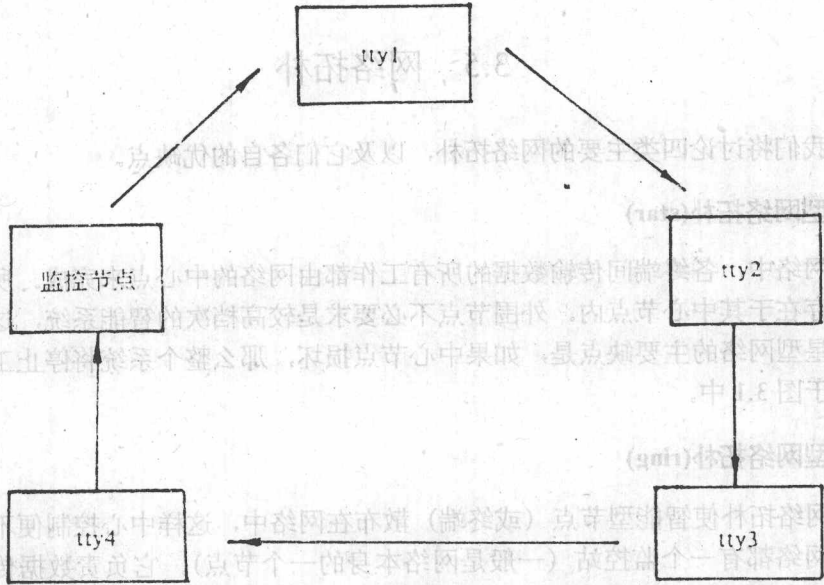


图 3.2 环型网络拓扑

当一个设备需要在网络上传输数据时，它首先“侦听”(listen)以确定网络是否忙（即是有数据正在传输）。当网络一旦空闲，该设备将传输它的信息。这种网络拓扑的优点在于网络的智慧存在于网络上各节点（终端）中。它们必须有识别地址的能力，以便它们可以接受数据，而且也要有能力确定什么时候发送信息是安全的（避免网络冲突）。这种类型的网络