

邓胜兰
等编



实用**UNIX**系统

教程

实用 UNIX 系统教程

邓胜兰 罗 军 吴庆波
夏卫民 罗 宇 尹俊文 编

国防科技大学出版社

图书在版编目(CIP)数据

实用 UNIX 系统教程/邓胜兰, 罗 军, 吴庆波, 夏卫民, 罗 宇, 尹俊文; —长沙: 国防科技大学出版社, 1997. 6

ISBN 7-81024-429-9

I 实用 UNIX 系统教程

Ⅱ 邓胜兰

Ⅲ 操作系统—教材

IV TP31

责任编辑: 文 慧

责任校对: 罗 青

封面设计: 陆荣斌

国防科技大学出版社出版发行

电话: (0731) 4555681 邮政编码: 410073

新华书店总店北京发行所经销

国防科技大学印刷厂印装

开本 787×1092mm 1/16 印张: 16.5 字数: 381 千

1997 年 6 月第 1 版第 1 次印刷 印数: 1—5000 册

ISBN 7-81024-429-9
TP·87 定价: 19.00 元

内 容 提 要

本书以UNIX系统V4.2为背景，面向初学者及系统开发人员，采用通俗易懂的语言，深入浅出地阐述了UNIX系统的基本概念、内部结构和工作原理，并且系统地介绍了UNIX的常用命令、实用工具、系统维护和网络管理。同时着眼于UNIX系统的最新发展动向，介绍了新的UNIX系统核心mach3.0，以及Internet、电子邮件等。本书的特点是将UNIX系统的工作原理与应用技术有机地结合在一起，融和了作者参加操作系统设计和开发的实践经验，注重理论与应用的结合，使读者全面了解和掌握UNIX操作系统。

前 言

UNIX系统是目前最流行的操作系统之一，几乎在所有类型的计算机上都配有UNIX系统。UNIX又是一个开放式系统，关于它的操作系统平台、API界面等已有国际上公认的标准，为其开发的应用程序、软件包应有尽有，包括了商业、科学计算及工程设计等各个领域。因此，掌握UNIX操作系统，将有利于开发和使用已在UNIX平台上积累的丰富的软件资源。

本书以UNIX系统V4.2为线索，阐述了UNIX操作系统的基本概念、工作原理、系统管理和应用技术。重点放在两方面：(1)着重介绍UNIX操作系统的基本概念、系统构成和实现机制，但不试图展开过多的理论上的探讨。(2)着重介绍UNIX系统管理和应用中一些常用的、重要的命令及软件工具的使用，并配有实例说明。本书将UNIX系统的工作原理与应用技术有机地结合在一起，引导读者注重理论与实践的结合，全面地掌握UNIX操作系统。

本书共分九章。

第一章概要介绍UNIX系统的产生和发展过程。

第二章阐述UNIX系统的基本原理和实现机制，包括进程管理、存储管理、文件系统、设备管理、进程间通信、SHELL机制，最后介绍了新的UNIX系统核心mach3.0。

第三章介绍UNIX系统的常用命令。

第四章介绍UNIX系统编辑软件vi和ed。

第五章介绍UNIX系统的shell。

第六章介绍UNIX系统的实用工具：make、Yacc、lex、sccs。

第七章介绍UNIX系统的维护和管理。

第八章介绍UNIX系统的网络管理及使用。

第九章介绍UNIX系统的电子邮件。

由于编者水平有限，书中难免有错误和不妥之处，恳切希望读者予以指正。

编者

1997.4于国防科技大学

目 录

第一章 UNIX操作系统概况

1.1 UNIX的产生	1
1.2 UNIX的发展	2
1.3 UNIX的特点	3
1.4 UNIX的未来发展	4

第二章 UNIX操作系统原理

2.1 基本概念	5
2.2 进程管理	7
2.2.1 进程的概念	7
2.2.2 进程上下文及切换	8
2.2.3 进程的状态与控制	10
2.2.4 进程的调度	12
2.2.5 软中断信号	13
2.2.6 系统自举和进程树	14
2.3 存储管理	15
2.3.1 页式 (paging) 存储管理	15
2.3.2 进程虚空间描述	17
2.3.3 进程虚空间操作	18
2.4 文件系统	19
2.4.1 文件系统结构	19
2.4.2 索引节点与目录	19
2.4.3 文件系统的空间分配	22
2.4.4 文件的打开和读写	23
2.4.5 管道	24
2.4.6 文件系统安装	25
2.5 设备管理	27
2.5.1 驱动程序接口	27
2.5.2 块设备I/O	28
2.5.3 字符设备I/O	29

2.5.4 流	30
2.6 进程间通信	32
2.6.1 进程跟踪	32
2.6.2 系统V IPC	33
2.6.3 套接字 (socket)	36
2.7 SHELL原理	39
2.7.1 shell 命令结构	39
2.7.2 shell 控制结构	41
2.7.3 shell 运行环境	42
2.7.4 其他shell	43
2.8 MACH: UNIX系统的新核心	44
2.8.1 MACH的基本概念	45
2.8.2 任务和线程	48
2.8.3 端口与IPC机制	51
2.8.4 虚存和存储管理	52
2.8.5 多处理机调度与管理	55
2.8.6 设备管理	58

第三章 UNIX操作命令

3.1 基本UNIX命令	60
3.1.1 注册进入系统 (login)	60
3.1.2 口令 (passwd)	60
3.1.3 显示日期和当前时间 (date)	61
3.1.4 查询当前登录在系统中的用户信息 (who)	61
3.1.5 查询当前登录在系统中的用户行为 (w)	61
3.1.6 向另外用户发送消息 (write)	62
3.1.7 msg命令 (msg)	62
3.1.8 向当前登录在系统中的所有用户发送消息 (wall)	62
3.1.9 退出系统 (logout)	62
3.2 UNIX文件系统的操作	63
3.2.1 目录管理	63
3.2.2 文件管理	65
3.3 其它UNIX命令	76
3.3.1 抽象字符的使用	76
3.3.2 输入/输出改向	77
3.3.3 管道	78
3.3.4 后台命令	79

3.3.5 查找字符串 (grep)	79
3.3.6 统计文件的行数、字数和字符数 (wc)	79
3.3.7 查看当前系统中的活跃进程 (ps)	80
3.3.8 终止指定进程 (kill)	80
3.3.9 软盘、磁带的使用 (format、tar、cpio、dd)	80
3.3.10 压缩和解压文件 (compress, uncompress)	82
3.4 UNIX系统下的有关DOS命令	82
3.4.1 dosformat	82
3.4.2 dosdir	83
3.4.3 dosmkdir	83
3.4.4 dosrmdir	84
3.4.5 doscat	84
3.4.6 doscp	84
3.4.7 dosrm	84

第四章 正文编辑器

4.1 ed编辑器	86
4.1.1 ed简介	86
4.1.2 ed的基本命令	86
4.2 全屏幕编辑器vi	95
4.2.1 用vi建立或编辑文件	95
4.2.2 vi的基本编辑命令	96
4.2.3 vi的高级命令	99

第五章 UNIX Shell

5.1 引言	105
5.2 输入/输出的转向	105
5.2.1 输出转向	106
5.2.2 输入转向	106
5.3 管道命令	106
5.4 执行和停止进程	107
5.4.1 用batch和at命令在以后某个时刻运行命令	107
5.4.2 终止活动进程	108
5.4.3 使用nohup命令	108
5.5 Shell内部命令	108
5.6 Shell编程	114

5.6.1 Shell变量	114
5.6.2 变量替换	116
5.6.3 Shell语句	117
5.6.4 Shell编程实例	121
5.6.5 Shell程序的调试	122
5.7 Shell的环境控制	124
5.7.1 终端特性设置	124
5.7.2 设置注册环境的文件. profile	124
5.8 cshell的环境控制	125
5.8.1 变量的设置	125
5.8.2 改变提示符设置	125
5.8.3 改变注册目录	125
5.8.4 改变查寻路径	126
5.8.5 设置历史表	126
5.8.6 改变命令的名字	127
5.8.7 关闭抽象字符的替代	127
5.8.8 禁止覆盖文件	127
5.8.9 改变终端类型	127
5.8.10 设置注册环境的文件	128
5.8.11 脱机处理文件	128

第六章 UNIX实用工具

6.1 awk	129
6.1.1 awk基本概貌	129
6.1.2 模式	131
6.1.3 动作	133
6.1.4 输出	136
6.1.5 输入	137
6.2 语言开发工具lex和yacc	138
6.2.1 lex和yacc的作用	138
6.2.2 lex源程序的格式	138
6.2.3 lex源程序的编写	140
6.2.4 yacc源程序的格式	141
6.2.5 yacc源程序的编写	143
6.2.6 lex和yacc之间的接口	146
6.3 C语言排错工具lint	146
6.3.1 使用lint的例子	147

6.3.2 类型检查	148
6.3.3 语句检查	149
6.3.4 lint库	150
6.3.5 lint命令行选项	150
6.4 make	150
6.4.1 make的基本功能	151
6.4.2 make的说明文件 (Makefile)	151
6.4.3 make的用法	152
6.4.4 make的内部规则	154
6.4.5 Makefile文件示例	155
6.5 SCCS	157
6.5.1 用sccs create产生SCCS历史文件	158
6.5.2 用sccs get提取当前版本	158
6.5.3 改动文件 (产生delta)	159
6.5.4 恢复旧版本	161
6.5.5 登记改动	161
6.5.6 速写记号	162
6.5.7 在工程项目中使用SCCS	162
6.5.8 保护措施	163
6.5.9 用sccs admin管理SCCS文件	163
6.5.10 维护不同的版本	164
6.6 调试工具debug	164
6.6.1 运行debug	164
6.6.2 debug的变量	165
6.6.3 debug的表达式	166
6.6.4 debug的进程控制	166
6.6.5 用户命令	167
6.6.6 一个debug的实例	172

第七章 系统维护与管理

7.1 系统控制与监视	175
7.1.1 成为超级用户	175
7.1.2 UNIX目录结构	176
7.1.3 系统启动与关闭	177
7.1.4 设置系统时钟	179
7.1.5 设置系统名	179
7.1.6 进程监视	180

7.1.7 检查系统配置	182
7.1.8 监视系统中的用户	183
7.2 用户管理	184
7.2.1 增加一个新的用户	184
7.2.2 删除用户	186
7.2.3 增加一个用户组	187
7.2.4 设置用户工作环境	188
7.2.5 环境变量	191
7.2.6 与用户通信	191
7.3 文件系统的管理与维护	193
7.3.1 文件系统的类型	193
7.3.2 建立文件系统	194
7.3.3 安装和拆卸文件系统	196
7.3.4 检查和修复文件系统	198
7.3.5 维护文件系统	199
7.4 软件包的安装	201
7.4.1 软件包的安装工具	201
7.4.2 软件包的安装方式	202
7.4.3 软件包安装前的准备	202
7.4.4 软件包的安装	204
7.4.5 安装正确性检测	206
7.4.6 显示软件包信息	207
7.4.7 软件包的删除	209
7.5 增加新的设备	210
7.5.1 增加打印机	210
7.5.2 增加新的终端	211
7.5.3 增加一个新硬盘	213
7.6 重新配置系统核心	216
7.6.1 重新配置系统参数	216
7.6.2 重新生成核心	218

第八章 网络管理及使用

8.1 网络管理	219
8.1.1 TCP/IP 协议	219
8.1.2 TCP/IP 网的安装	221
8.1.3 网络故障检测	226
8.2 UNIX网络操作	230

8.2.1 显示局域网中各计算机状态·····	230
8.2.2 显示网络中的用户信息·····	230
8.2.3 远程登录·····	232
8.2.4 远程文件拷贝·····	234
8.2.5 ftp的使用·····	235
8.2.6 NFS的使用·····	237

第九章 电子邮件

9.1 mail命令·····	240
9.1.1 发送电子邮件·····	240
9.1.2 处理电子邮件·····	243
9.1.3 转发电子邮件·····	245
9.2 mailx命令·····	246
9.2.1 发送邮件·····	246
9.2.2 邮件处理·····	248

第一章 UNIX操作系统概况

UNIX是目前世界上最“古老”的也是最流行的操作系统之一。说它“古老”，是因为相对于其它操作系统而言，它的历史悠久。几乎在所有计算机厂家、大专院校及研究所里，都可见到它的身影。回顾一下UNIX的产生和发展历史，可以帮助我们了解它具有如此强大生命力的原因所在，把握它的设计精神、功能特点和发展方向。

1.1 UNIX的产生

1965年，美国贝尔实验室和通用电气公司及麻省理工学院的MAC课题组一起联合开发一个被称为Multics的新操作系统。Multics系统的目标是要向大的用户团体提供对计算机的同时访问，支持强大的计算能力与数据存储，以及允许用户在需要的时候容易地共享他们的数据。到1969年，Multics系统已在GE645计算机上运行了，但是令人遗憾的是，Multics未能达到它预期的目标，结果此项目最终被放弃了。

当时在贝尔实验室工作的Ken Thompson 和Dennis Ritchie也参加了Multics的工作，从中获得一些有益的经验。Multics的工作结束后，Ken Thompson 和Dennis Ritchie及其他人为改善他们的程序设计环境，设想了一个新的文件系统，它后来就演化为UNIX文件系统的早期版本。Thompson编写了若干程序，在GE645上模拟实现所设想的文件系统行为。与此同时，Thompson还在GE645上编写了一个名为“宇宙旅行”的游戏程序，他发现此程序运行时开销太大，并且很难控制“宇宙飞船”。于是，他找来一台几乎无人问津的但却能提供良好的图形显示和开销廉价的PDP-7计算机，打算将“宇宙飞船”游戏程序移植到此机器上。移植工作并不像想象中的那样简单，Thompson和Ritchie决定为PDP-7实现一个简单的操作系统，其中包括那个新设想的文件系统（使用层次目录结构和inode数据结构），支持两个用户分时共享和内存管理的进程子系统以及一部分实用程序。此操作系统是用汇编语言写的，也就是最早的UNIX的汇编版本。

UNIX系统的第一位真正用户是贝尔实验室的专利部门，该部门要求在PDP 11/20计算机上建立一个功能很强的文字处理系统。于是，第一个UNIX汇编版本于1971年被移植到PDP 11/20上，成为UNIX操作系统第一版。它支持更多的用户分时共享系统资源，并带有一个功能很强的文字处理系统roff。与此同时，Thompson在编写PDP 11/20的FORTRAN编译器的过程中，发明了一种新的低级语言B，在Ritchie的帮助下，Thompson对语言B进行了改造，研制出C语言及其编译器。C语言以高级程序设计语言的结构和编程环境，提供了类似汇编语言那样的系统资源操纵能力和程序的执行效率。1972年，Thompson和Ritchie用C语言重写了UNIX操作系统（第二版）。重写后的UNIX操作系统不仅保持了原有的执行效率，而且能比较容易地移植到其他支持C编译器的机器上。

1.2 UNIX的发展

1974年, Thompson和Ritchie在《ACM通讯》上发表了一篇描述UNIX系统的文章, 第一次向公众介绍UNIX系统, 在计算机学术界引起强烈反响。贝尔实验室还将UNIX系统源码免费提供给一些大学、研究机构用于教育和研究, 许多学者、研究人员对此表现出极大的兴趣, 他们在UNIX系统上编写了大量的应用程序, 并试图对UNIX系统进行改造并增强其功能。到1977年, UNIX系统的装机数目增长到大约500个, 其中20%在大学。UNIX系统开始在电话公司流行起来, 为程序开发、网络事务操作服务及实时服务提供了良好的环境。这一切都使UNIX系统的声望日益增长。

1977年, AT&T开始将UNIX系统的许可证提供给商业机构。同年, 美国交互系统公司(Interactive Systems Corporation)成了UNIX系统的第一个增值转卖商, 他们增强了它的功能, 使之用于办公室自动化环境中。同样, 1977年也是标志UNIX系统首次被移植到非PDP计算机(Interdata 8/32计算机)上的一年。从那时起, 许多计算机厂商纷纷向AT&T购买UNIX系统的许可证, 将其移植到自己的机器上, 并根据计算环境的需要, 作功能上的改造和增强, 进而推出自己的以UNIX为基础的操作系统。例如, IBM公司的AIX, 苹果公司的A/UX, HP公司的HP/UX, SUN公司的Sun OS, Microsoft公司的XENIX, Intel公司的XENIX 286, SCO公司的SCO XENIX 386等等。与此同时, AT&T继续UNIX系统的研究工作, 不断推出新版本。1989年, AT&T宣布了它的新版本UNIX System V Release 4 (简称为SVR4)。

UNIX系统的各种变体的出现, 一方面表明了UNIX系统的普及和发展, 另一方面这些变体之间的不兼容, 给软件开发者和用户带来许多不便, 造成投资的重复和浪费。因此, 不管是计算机厂商还是计算机用户, 都需要制定某种标准来规定应用程序设计界面(API)及其它界面, 以保护各自的投资利益。目前已被广泛接受的UNIX标准有:

- POSIX 关于计算环境的可移植操作系统界面标准;
- SVID 由UI (UNIX INTERNATIONAL, INC.) 组织中各成员公司联合制定的关于SVR4的标准;
- X/OPEN 将国际上已公认的和事实上的标准结合起来, 形成一个关于开放的、交互的计算机应用环境标准XPG。

前面所述的提供类UNIX操作系统的公司, 都宣布自己的产品符合上述一个或多个标准。

进入80年代后, UNIX系统及其应用软件发展迅速, 市场日益扩大, 功能不断加强和完善。世界上几家主要的计算机公司都推出自己的类UNIX的操作系统作为主要产品或替代产品, 例如CRAY公司为其巨型机配备的UNICOS。迄今为止, 没有一种操作系统能够像UNIX一样适应如此广泛的硬件范围, 从微型机、小型机、中型机到巨型机。

所有的UNIX操作系统可划分成三大类:

- 基于AT&T System V UNIX系统V是AT&T研制的。AT&T拥有其版权, 并向商业机构提供许可证。DEC、IBM等大公司均购买了其许可证。

· 基于BSD 在美国国防部的资助下, 加州大学伯克利分校的一个研究小组研制的一个UNIX系统, 称作BSD。目前, UNIX系统中广泛采用的虚存管理、全屏幕编辑、Socket机制和网络功能等首先是在BSD中实现的。如果说AT&T UNIX适合商业事务性处理, 那么BSD更适合于软件研究开发。目前BSD已发展到BSD 4.3。

· 基于XENIX XENIX是Microsoft公司为Intel 一族的微型机研制的类UNIX系统。SCO公司购买了其版权, 研制出自己的XENIX 386。

1.3 UNIX的特点

如果用一句话来描述UNIX系统的特点, 那就是功能简洁、层次清晰。这主要是由最初研制它的人员环境所导致的。Thompson和Ritchie当时研制这个系统不是为开发商品, 而是为了自己使用, 是想为自己建立一个舒适的软件开发环境。因此, 设计时力求简单实用, 这样设计出来的操作系统就非常简洁。由此带来的结果是, UNIX系统具有许多引人注目的特点:

· 该系统以高级语言书写, 易读、易懂、易修改, 易移植到其他机器上。据Ritchie估计, 用C语言书写的第一个系统与用汇编语言书写的系统相比, 大约慢20%~40%。但是, 采用高级语言的优点要远远超过它的缺点。现在研制的操作系统大都采用C语言编写程序。

· 它提供简单的、基本的系统调用界面和一个功能很强的命令处理器shell, 支持一组能够由较简单的程序构造出复杂程序的原语, 如管道操作、I/O改向等。这使得软件开发人员能够方便地编写高效率的应用程序。UNIX系统的许多高级功能都是以实用程序的形式实现的。

· 它使用了易于维护、实现高效的层次结构文件系统。文件采用字符流这样的一致格式, 使应用程序易于编程和处理。

· 它是一个多用户、多进程系统。目前已开发出UNIX多机、多线程版本;

· 支持多种文件系统, 例如, s5、ufs等;

· 重要的是, 它是一个开放系统, 已有公认的API标准。用户可以方便地编写UNIX应用程序, 而不必考虑具体的机器硬件结构。

UNIX系统也有一些不足之处, 例如:

· UNIX的用户界面不太友好。对一般用户而言它有点难以掌握, 它更适合软件开发人员的口味。现在, UNIX系统上的图形界面系统正努力弥补这一不足之处;

· UNIX不太适合实时处理, 因为它只能在预定的几个“安全点”作进程切换来响应实时处理。

AT&T的UNIX系统USVR4中实现的主要功能有:

· TCP/IP协议;

· C shell、kern shell;

· 套接字(sockets)机制;

· 作业控制;

- 符号链接;
- 增强的信号处理;
- Berkeley快速文件系统 (UFS);
- 虚拟文件系统 (VFS);
- 远程文件系统 (RFS);
- 进程间通信 (IPC);
- 远程过程调用 (RPC);
- vnode数据结构;
- Open look窗口系统;
- 内存映射文件;
- 外部数据表示 (XDR);
- 共享库;
- 流 (STREAMS) 通信机制;
- 实时支持。

1.4 UNIX的未来发展

目前, UNIX系统发展的进入了一个新的重要时期。新的软件技术、网络技术及安全技术不断引进UNIX系统, UNIX系统及其变体的新版本纷纷出台, 令人眼花缭乱。例如, USL与Novell公司共同开发的USVR4的桌面系统版本Unixware。Unixware支持更强的网络功能, 提供更友好的用户窗口界面。

UNIX虽然是一个易移植、易扩充的系统, 但是, 多年来人们不断地在UNIX系统内核中添加新的功能和特性, 以满足计算环境的需要, 使原来那个比较小、比较紧凑的内核逐渐变得庞大而复杂, 难以维护。越来越多的人开始意识到这点, 并寻求解决问题的途径。1984年卡内基梅隆大学采用微内核 (Microkernel) 技术研制出一个新的操作系统核心Mach, 就是一个成功的范例。

Mach的设计思想是, 核心只提供必要的几个简单的用于建立复杂操作的对象, 将尽量多的传统操作系统的功能模块放到核心外的用户层实现, 而这一点正符合UNIX小而精的初衷。在Mach 3.0中, UNIX是作为核心外的一个应用程序UNIX server的身份实现的, 可提供与BSD4.3二进制兼容的应用环境。Mach具有良好的可移植性、可调性和可扩充性, 可支持松散型及紧密藕合型多处理机体系结构, 提供良好的分布及并行计算环境。

OSF (Open Software Foundation) 在它推出的OSF/1操作系统中采用了Mach作为核心。OSF/1在安全性、多处理、多线程及其它特性都有所加强的Mach基础上, 提供BSD的内核功能, 能很好地支持多处理环境和多线程并发, 并遵从所有有关的UNIX标准和规范。UI (UNIX International) 也表示, 将在它的新UNIX多机版本中吸收Mach的优点及采用微内核技术。除Mach 3.0外, 还有Chorus 和Amoweba系统亦采用了微内核技术。Mach3.0在国际上已成为十分著名的多机操作系统, 它对UNIX的未来发展将起着重要作用。

第二章 UNIX操作系统原理

2.1 基本概念

概括地说，操作系统主要有两方面的功能：一是给用户提供一个方便而且强有力的使用计算机资源的环境；二是充分、合理地分配使用系统内包含的各种软硬件资源，提高整个系统的使用效率和经济效益。在操作系统的支持下，用户可以访问远程文件而不必关心网络操作细节，用户可以分时共享CPU而不会感觉到他人的使用。

文件和进程是UNIX系统中两个最基本的概念。文件是具有名字的有序的数据集合。例如，一个C或FORTRAN源程序、一个目标代码程序、系统中的库程序、一批待加工处理的数据、一篇文章等都可构成一个文件。进程是程序的一次执行，是系统进行资源分配和调度运行的一个独立单位。一个进程循着一个严格的指令序列执行，这个指令序列是自包含的，并且不会跳转到别的进程的指令序列上。它读写自己的数据和栈区，但不能读写其他进程的数据和栈区。进程可以通过系统调用与其他进程及外界进行通信。核心为每个进程建立一张资源登记表，记录它使用的各种资源，如内存、文件等。

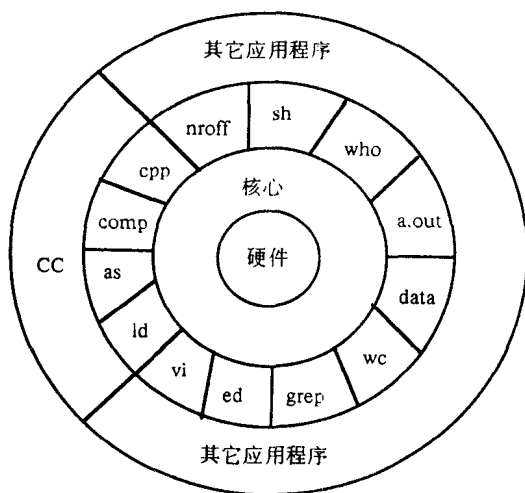


图2.1 UNIX系统的体系结构

UNIX系统的体系结构示于图2.1。UNIX核心直接与硬件交互，向外提供UNIX系统调用界面。外层的程序，诸如shell及编辑程序(ed与vi),通过引用系统调用请求核心完成各种操作，并在核心与调用程序之间交换数据。其他应用程序能在较低层的程序或工具的基础