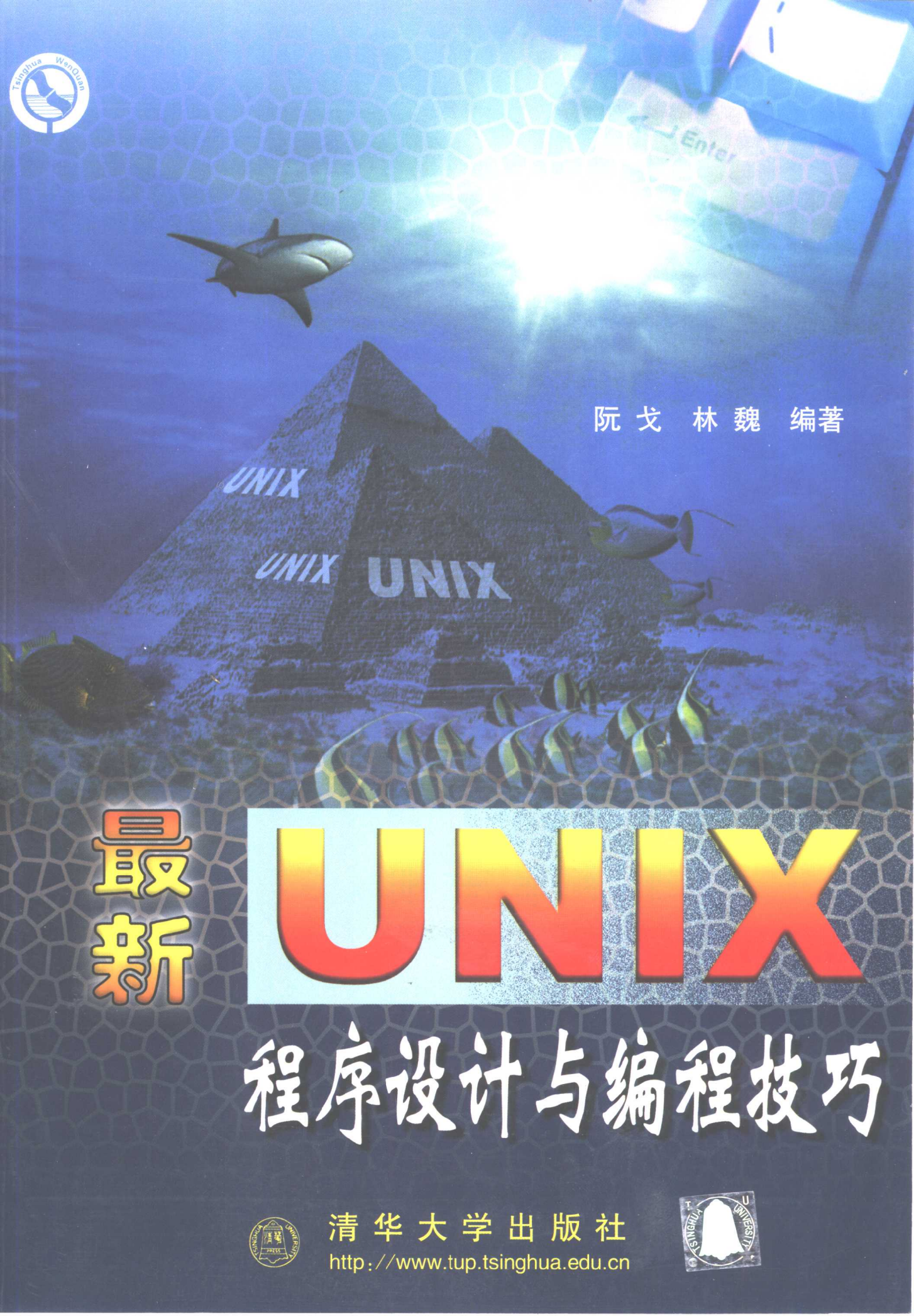




阮戈 林魏 编著

最新

UNIX

程序设计与编程技巧



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



最新 Unix 程序设计与编程技巧

阮戈 林巍 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书从实用的角度出发,对 Unix 环境下的 C 语言编程作了较为全面的分析和介绍。全书分为三部分。第一部分介绍了 Unix 的基本知识、在 Unix 下编译工具的使用和 shell 编程。第二部分详尽而深入地介绍了在 Unix 下的系统高级编程,包括 Unix 的文件子系统、进程控制、进程间通信和 BSD Socket 编程,这些也是本书的核心内容。第三部分则从提高开发效率角度出发,简要介绍在 Unix 环境下另外几种最流行的语言——Perl、PHP 和 Java 的核心内容。

书中穿插给出了丰富的实例程序以辅助讲解,而且其中相当部分属于经典例程,不乏可圈可点之处,对于提高读者的编程能力颇有裨益。

本书内容覆盖面广而深入,适用于有一定 C 编程经验而想转向 Unix 环境下进行开发的人员,以及 Unix 环境编程爱好者进行自学,同时也可以作为广大 Unix 系统程序管理和开发及维护人员的参考用书。

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,无防伪标签者不得销售。

书 名: 最新 Unix 程序设计与编程技巧

作 者: 阮戈 林巍

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编: 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 章忆文 钱新

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 30.75 字数: 744 千字

版 次: 2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷

书 号: ISBN 7-302-04500-3/TP·2660

印 数: 0001~5000

定 价: 46.00 元

前 言

Unix 是一种非常重要的操作系统。对于普通个人用户来说, 它显然没有微软的 Windows 系统那样流行, 但是作为一种主要运行于服务器和工作站上的操作系统, 无论是工业应用、商业应用或科学计算, Unix 都以它的高稳定性、强大的功能深得广大用户的信赖, 牢牢占据着服务器的高端市场。在这种形势下, 人才市场对 Unix 环境下的开发人员的需求始终有增无减。但是目前国内相关的中文书籍寥寥无几, 基本上是个空缺, 这无疑又加高了 Unix 编程的门槛。出于这些考虑, 我们精心编写了此书。

特别需要指出的是, 此书不仅适于有志于在 Unix 下进行开发的人员, 而且对于有兴趣于 Unix 的系统管理(如果不了解 Unix 下编程原理, 就不是一个称职的系统管理员)和系统安全的人员来说也具有参考价值。

本书按内容分为三个部分:

第一部分介绍的是 Unix 操作系统的基础知识。通过这一部分的讲解, 使编程用户逐渐了解他们即将面对一个什么样的环境, 这些环境能够为他们的编程工作提供什么样的支持。比如: 文本编辑工具的使用、源程序的编译方法以及程序的调试等等。这部分针对的对象是 Unix 的入门级用户。

第二部分是本书的核心部分。该部分结合 Unix 操作系统原理, 对 Unix 环境下的 C 高级编程进行深入浅出的分析。对于那些已经具备其他工作平台的编程能力, 并且对操作系统原理有所了解的读者, 学完此部分后, 无论是编程实践还是操作系统理论, 均能得到不同程度的提高。这部分适宜的对象是 Unix 开发的中高级用户。对于初学者, 此部分可能稍难, 学习时不能浮躁。另外, 由于编撰体系的需要, 有些概念对于初学者而言, 超前出现, 不理解之处, 可以暂时跳过, 等到读完后面相关的章节就可迎刃而解。

第三部分简要介绍其他几种在 Unix 操作系统环境下常用的开发语言。比如: Perl, PHP 和 Java。它们都是当前比较流行的。读者也可以把这一部分当作这几种语言的入门教程。这部分的目的是开阅读者思路, 在实际开发时, 可以根据需要灵活选择不同的语言, 以减少开发的任务量。

本书在写作过程中参阅了大量的资料, 以最大程度地保证概念的准确性。书中的例证丰富, 且大多数例子都经过精心筛选, 有的用来提示编程注意事项, 有的则演示程序设计技巧。

本书编撰分工如下: 第二部分编写工作由阮戈完成, 其他部分的编写由林巍完成。由于作者水平有限, 难免有疏漏和谬误之处, 敬请读者批评指正。

2001 年 4 月

北 京

目 录

第一部分 Unix 编程基础

第 1 章 Unix 操作系统概述.....	1
1.1 Unix 系统的历史.....	2
1.1.1 Unix 的产生与发展.....	2
1.1.2 Unix 的分支.....	2
小结.....	3
1.2 Unix 操作系统的用户基础.....	4
1.2.1 操作系统.....	4
1.2.2 初次使用 Unix 操作系统.....	4
1.2.3 文件系统.....	6
1.2.4 程序和进程.....	8
1.2.5 Unix 系统管理.....	10
小结.....	11
第 2 章 Unix 下的编程环境.....	12
2.1 vi 的使用.....	13
2.1.1 概述.....	13
2.1.2 vi 的命令.....	14
小结.....	16
2.2 编译器和调试工具.....	16
2.2.1 编译器.....	17
2.2.2 调试工具.....	19
小结.....	27
2.3 make 工具及 makefile 规则.....	28
2.3.1 make 程序概述.....	28
2.3.2 makefile 内部规则.....	33
小结.....	34
第 3 章 shell 简介及 shell 编程.....	35
3.1 shell 概述.....	36
3.2 shell 的标准输入/输出及重定向.....	37
3.2.1 shell 的标准输入/输出.....	37
3.2.2 输入/输出重定向.....	37

3.2.3 管道线	39
3.3 进程控制	40
3.4 通配符、特殊字符和转义字符	41
3.5 shell 程序脚本设计	43
3.5.1 shell 程序脚本的基本结构	43
3.5.2 如何执行 shell 程序	43
3.5.3 shell 变量	44
3.5.4 条件测试	46
3.5.5 控制流结构	50
3.5.6 shell 函数	53
3.5.7 脚本的调试	53
小结	54
第 4 章 Unix 下几种重要的工具	55
4.1 正则表达式	56
4.1.1 概述	56
4.1.2 正则表达式的使用	56
小结	60
4.2 模式搜索程序 grep	61
4.2.1 怎样使用 grep	61
4.2.2 行匹配	62
4.2.3 精确匹配	62
4.2.4 正则表达式匹配	63
小结	64
4.3 流编辑程序 sed	64
4.3.1 怎样使用 sed	64
4.3.2 怎样通过 sed 查询文本	65
4.3.3 怎样通过 sed 编辑文本	65
小结	69
4.4 非常实用的数据加工和检索工具 awk	69
4.4.1 基本概念和基础操作	70
4.4.2 其他重要概念	72
4.4.3 应用于 awk 编程脚本的动作流程控制	75
4.4.4 与其他命令及 shell 一起使用 awk	78
小结	78
 第二部分 Unix 下的 C 语言高级编程	
第 5 章 Unix 系统结构	79

5.1	Unix 层次结构.....	80
5.2	Unix 内核特点.....	81
5.3	系统调用和库函数	81
第 6 章	文件子系统	82
6.1	文件子系统的基本概念	83
6.1.1	超级块	83
6.1.2	索引节点	84
6.1.3	文件的多重索引结构	85
6.1.4	目录文件	86
6.1.5	索引节点和磁盘分配与释放	87
	小结.....	88
6.2	文件的基本操作.....	89
6.2.1	文件描述符	89
6.2.2	文件的创建和删除	89
6.2.3	文件的打开和关闭	94
6.2.4	文件的读/写操作	97
6.2.5	文件的随机存取	100
6.2.6	综合应用实例	102
	小结.....	105
6.3	文件高级操作.....	105
6.3.1	文件的模式	106
6.3.2	文件控制	109
6.3.3	目录文件操作	122
6.3.4	文件信息查询	129
6.3.5	链接——Link 和 Symlink 系统调用	133
6.3.6	综合应用实例	136
	小结.....	141
6.4	设备文件操作.....	141
6.4.1	设备文件基础知识	141
6.4.2	两个系统调用——mknod 和 ioctl	143
6.4.3	终端设备操作	144
6.4.4	设备驱动程序的编写	152
6.4.5	硬、软盘设备操作	175
	小结.....	180
第 7 章	进程控制	181
7.1	进程的基本概念.....	182
7.1.1	进程的结构	182

7.1.2 进程状态及其转换	184
7.1.3 系统调度简介	185
小结	186
7.2 进程的控制	186
7.2.1 进程的建立——fork 系统调用	186
7.2.2 执行新程序——exec 系统调用	188
7.2.3 进程的终止——exit 系统调用	191
7.2.4 进程的同步——wait 系统调用	192
7.2.5 悬挂进程——sleep 系统调用	195
7.2.6 进程控制的综合实例	195
小结	202
7.3 进程的高级操作	202
7.3.1 进程标识号管理	202
7.3.2 进程的用户标识号管理	204
7.3.3 进程的环境管理	205
小结	207
7.4 守护进程	207
7.4.1 守护进程及其特性	207
7.4.2 守护进程的编程规则	208
7.4.3 守护进程实例	210
7.4.4 守护进程的错误输出	211
小结	213
第 8 章 进程间的通信	214
8.1 IPC 一瞥	215
8.2 信号	216
8.2.1 信号的基本知识	216
8.2.2 信号类型	218
8.2.3 内核对信号的处理	220
8.2.4 信号的基本操作	222
8.2.5 信号的其他操作	231
8.2.6 信号的阻塞	236
小结	240
8.3 管道和 FIFO	240
8.3.1 管道	240
8.3.2 有名管道——FIFO	248
8.3.3 字符流的分隔	255
小结	256
8.4 System V 消息队列	256

8.4.1	System V IPC	256
8.4.2	消息队列基本概念	258
8.4.3	操作消息队列的系统调用	260
8.4.4	迭代客户-服务器实例	265
8.4.5	并发客户-服务器实例	269
	小结	272
8.5	信号量	273
8.5.1	信号量的基本概念	273
8.5.2	信号量的操作	274
8.5.3	一个实例 semtool	276
8.5.4	利用信号量实现对记录的上锁	282
	小结	284
8.6	共享内存段	284
8.6.1	共享内存段的基本概念	284
8.6.2	共享内存段的操作	285
8.6.3	共享内存段应用实例	287
	小结	289
第 9 章	Socket 网络编程	290
9.1	网络基础知识	291
9.1.1	计算机网络的概念	291
9.1.2	网络分层结构	291
9.1.3	TCP/IP 协议集	292
	小结	295
9.2	基本 Socket 编程	296
9.2.1	套接口 Socket 概念简介	296
9.2.2	套接口 Socket 使用简介	297
9.2.3	套接口地址结构	298
9.2.4	IP 地址和如何使用 IP 地址	299
9.2.5	套接口编程的基本系统调用	300
9.2.6	简单 Client-Server 实例	307
9.2.7	I/O 多路复用——select 调用	314
9.2.8	套接口选项	316
9.2.9	TCP 的 Proxy 实例	322
9.2.10	非阻塞 I/O	329
9.2.11	套接口超时	341
	小结	343
9.3	高级 Socket 编程	343
9.3.1	inetd 守护进程	343

9.3.2 带外数据	348
9.3.3 线程简介	353
9.3.4 原始套接口	358

第三部分 Unix 下的其他开发语言

第 10 章 Perl 编程简介	378
10.1 Perl 的简介	379
小结	380
10.2 标量及其运算	380
10.2.1 标量	380
10.2.2 标量的运算	382
小结	384
10.3 Perl 的语法	384
小结	385
10.4 列表、数组和关联数组	385
10.4.1 列表	385
10.4.2 列表数组	385
10.4.3 关联数组	386
小结	390
10.5 子程序	390
小结	392
10.6 文件以及文件系统的操作	392
10.6.1 打开和关闭文件	393
10.6.2 读文件和写文件	393
10.6.3 命令行参数的运用	393
10.6.4 判断文件状态	394
10.6.5 文件系统的操作	395
小结	397
10.7 Perl 的流控制	397
10.7.1 条件判断	397
10.7.2 循环语句	398
小结	401
10.8 Perl 的指针——引用	401
10.8.1 引用的概念	401
10.8.2 对简单标量变量的引用	402
10.8.3 对列表数组的引用	403
10.8.4 关联数组的引用	404

10.8.5 参数的引用	406
10.8.6 子程序的引用	407
10.8.7 文件句柄的引用	408
小结	408
10.9 Perl 与 Unix 系统的互动操作	409
小结	410
第 11 章 PHP3 编程简介	411
11.1 PHP 的简介	412
11.1.1 PHP 的诞生与发展	412
11.1.2 PHP 的特点	413
11.1.3 PHP 的编译	413
小结	414
11.2 PHP 的变量	414
11.2.1 内部变量	415
11.2.2 动态变量	417
11.2.3 外部变量	417
小结	418
11.3 变量类型的转换	418
11.3.1 强制类型转换	419
11.3.2 字符串转换	419
小结	419
11.4 PHP 的运算符	420
11.4.1 三重运算符	420
11.4.2 算术运算符	420
11.4.3 条件运算符	420
11.4.4 一元运算符	421
11.4.5 位运算符	421
11.4.6 字符串连接运算符	422
11.4.7 逻辑运算符	422
11.4.8 赋值运算符	424
小结	425
11.5 PHP 的程序结构	425
11.5.1 表达式	425
11.5.2 语句	426
11.5.3 函数	426
小结	428
11.6 PHP 的类	428

小结.....	430
第 12 章 Java 编程	431
12.1 Java 的产生与发展	432
12.1.1 Java 的历史.....	432
12.1.2 Java 的特点.....	432
12.1.3 Java 的应用.....	433
小结.....	434
12.2 Java 程序的编译、调试和运行	434
12.2.1 JDK 简介	434
12.2.2 如何编译和调试 Java 源程序	434
12.2.3 如何执行 Java 源程序.....	435
小结.....	436
12.3 Java 数据类型及运算	436
12.3.1 Java 的数据类型.....	436
12.3.2 数据类型转换.....	438
12.3.3 数据运算符.....	439
小结.....	443
12.4 Java 数组.....	443
12.4.1 一维数组.....	443
12.4.2 多维数组.....	446
小结.....	449
12.5 Java 的对象、类、包和接口	449
12.5.1 对象、类、方法和消息之间的关系	449
12.5.2 继承和多态性.....	450
12.5.3 Java 的类.....	451
12.5.4 Java 的对象.....	456
12.5.5 Java 的包.....	458
12.5.6 Java 的接口.....	460
12.5.7 完整的 Java 源文件.....	462
小结.....	463
12.6 Java 的程序流控制	463
12.6.1 分支语句.....	463
12.6.2 循环语句.....	467
12.6.3 异常事件处理语句.....	470
12.6.4 注释语句.....	471
小结.....	471
12.7 Java 的线程控制	471

12.7.1 线程.....	471
12.7.2 构造一个线程体.....	473
12.7.3 同步与共享.....	475
小结.....	476

第一部分 Unix 编程基础

第 1 章 Unix 操作系统概述

本章要点:

Unix 是世界上最流行的操作系统之一。通过学习 Unix 操作系统的发展历程, 可以使我们了解为什么这样一个小型操作系统会占据了工作站操作系统的绝大部分市场, 并开始蚕食微型机、大型机乃至巨型机操作系统市场, 而风靡全球, 会具有如此强大的生命力。此外, 作为一个即将从事 Unix 编程的程序人员, 对 Unix 操作系统也应该先有一个粗浅的了解。

本章具体包括以下内容:

- ◆ Unix 的产生和发展
- ◆ Unix 使用简介

1.1 Unix 系统的历史

1.1.1 Unix 的产生与发展

Unix 是世界上最流行的操作系统之一。

概括来讲, Unix 操作系统是一种分时、多用户和多任务的操作系统。它具有很好的可移植性、可靠性、抗毁性和一致性, 并在几乎所有的硬件平台上都有其相对应的版本。Unix 产生于 20 世纪 60 年代末。它最初的设计思想来自于 AT&T Bell 实验室、MIT 和 GE 联合开发的 Multics 系统。Multics 并不是第一个交互式的多用户分时操作系统。在此之前, 欧洲已经开发出剑桥多路存取系统(Cambridge Multiple Access System), 而另一个兼容分时系统(Compatible Time Sharing System)也已经广泛应用。Multics 系统的最主要特点是: 它是一个分时交互式的操作系统。它的设计者 D. Ritchie 和 K. Thompson 为了改善程序设计环境又设计出了一个简单文件系统和操作系统内核的雏形。在当时, 他们同时还在设计一个太空旅行的游戏软件。为了使这个软件能够在当时的一种 PDP-7 计算机上顺利运行, 他们继续对这个系统雏形进行完善, 最终使一种新的操作系统在 PDP-7 上运行起来了。这种系统吸收了 Multics 系统的精华, 并做了许多简化。为了体现这种继承关系, 新系统被称为 UNICS, 后改名为 Unix。

Unix 操作系统的出现, 并不是出于某种商业目的, 而是直接来自编程用户的需要, 所以具有非常好的编程环境。

此后, Unix 操作系统得到逐步完善, D. Ritchie 和 K. Thompson 写出了它的用户手册, Unix 第一版诞生了。在这个版本中, Unix 的主体思想已经初见端倪, 例如文件系统、进程管理、系统界面以及一些至今仍在应用的命令等。

在 Unix 第二版问世以后, K. Thompson 又开发出了 B 语言。D. Ritchie 将 B 语言发展成为 C 语言。1973 年, 他们用 C 语言重新写了 Unix 第三版。由此可见, Unix 与 C 语言有着千丝万缕的联系。因此 Unix 操作系统作为 C 语言的编程环境可以使其功能发挥到极至。

此时的 Unix 操作系统与现在的系统已经非常相似了。

1.1.2 Unix 的分支

Unix 的进一步发展是在大学院校里完成的。在 Unix 开始得到广泛应用时, Bell 实验室低价出售了一批 Unix 操作系统给一些大学, 使得这些学校的学生逐渐熟悉这种非常便于程序开发的操作系统环境。等到这些学生毕业以后走向社会, 他们自然希望仍然有这样一个熟悉的系统环境, 这就导致 Unix 系统得以迅速走向社会和工业领域。

讲到这里, 首先应该提到的是 BSD Unix 系统。它是加州大学伯克利分校计算机科学系对 Unix 操作系统的改进版本。BSD 是 Berkeley Software Distribution 的英文首字母缩写。

它的特点是适合用来进行软件系统的开发。它的成长本身恰恰体现了大学里的学生群体对 Unix 操作系统的贡献。

另外一个比较重要的版本是 Unix System V, 它是由 AT&T 公司推出的商业版本的 System III 的改进版本。System V 加强了文件系统, 改进了终端驱动程序, 推广了进程通信, 实现了内存共享及信号量等重要功能。

这两种版本算是 Unix 系统的两大最主要的分支。BSD 大多应用于科研院校, 而 System V 则常见于工商企业。此外, 许多公司看重 Unix 的强大生命力, 也在开发自己的 Unix 版本。例如, 微软和 SCO 联合开发的 XENIX 在微机市场上就占有重要地位。

Unix 操作系统具有比较鲜明的特点。除了传统操作系统所常见的文件管理系统和进程管理系统以外, Unix 还具有强大的网络功能。这对于目前网络发展铺天盖地的世界来说, 可以说是最大的卖点。最近 Linux 的迅速成长就是最显然的例证。Unix 操作系统的多用户管理系统、分时系统以及良好的安全性能, 使得网络用户既能够方便地共享资源, 又能够得到安全的数据资源的保护。

Unix 的另外一个特点是使用非常方便。用户可以通过简单的命令行来完成大量的工作, 也可以通过简单的 shell 程序设计完成计算机的运算时间自动分配, 而无需时时干预, 节省了大量人力资源。每个使用过 Unix 操作系统的人都会为它简单而强大的功能所叹服。

Unix 几乎可以在已有的所有平台上运行。许多厂商购买了其源代码, 并开发出他们自己的版本。目前, 在微机系统上运行的 Unix 主要有 SCO XENIX, SCO Unix, Unix WARE 和 Solaris x86 等。在小型机上有 IBM AIX, HP UX 和 SUN Solaris 等。目前最流行的是 Linux 操作系统。

Linux 是一种由芬兰人 Linus Benedict Torvalds 于 1991 年设计的, 可以应用于个人电脑上的 Unix 操作系统。它不仅继承了 Unix 的优点, 而且比 Unix 更加开放。Linux 的源代码是完全免费公开的, 包括系统核心、驱动程序、开发工具包和应用程序。它的源代码被放在互联网上供用户随意下载和改进, 而设计者本人也非常欢迎人们对 Linux 进行开发和改进。这样做不仅对整个操作系统的完善和除错非常有益, 而且对广大计算机底层技术的爱好者来说也是一大福音。他们可以了解到系统是怎样运行的。就像市面上有一种非常别致的透明手表一样, 你可以看见齿轮机械的转动。这种感觉会让人们有一种探求其中奥妙的冲动。目前 Linux 有好几种发行套件, 比较多见的有 RedHat, Slackware 和 Debia 等。国内也出现了诸如红旗 Linux 和幸福 Linux 等发行套件。

对于那些厌烦了又大、又乱、又昂贵的 Microsoft Windows 的人, 或者那些想深入了解操作系统内部的奥秘, 想学习一种比较先进的操作系统, 但是口袋里银子又不够多的计算机爱好者来说, Linux 可以说是最值得推荐的一种选择。

小结

本节介绍了 Unix 的发展历史以及它的分类。作为一个即将使用 Unix 编程平台的程序员来说, 这些知识虽然还过于简单, 但是这种对 Unix 操作系统背景知识的了解将会增加用户对这种操作系统的热望和兴趣。

1.2 Unix 操作系统的用户基础

本节将给初次使用 Unix 操作系统的用户作一些初步的介绍,以便使用户对操作系统有一个整体的认识,其中有些内容还会在后面的章节给出更加详细的描述。

首先,对于一个一直就处在某一种操作系统如 Windows 或 MS-DOS 环境下工作的用户来讲,对什么是操作系统,它有什么用,与其他操作系统相比较有什么优缺点这样的一些问题恐怕就不太清楚,这是由于他们所需要做的就是使用好当前的操作系统。因此,一旦接触到一个新的、有所不同的操作系统时就会很自然地产生这样那样的疑问:我们为什么要使用这种操作系统,它比原来的操作系统有什么好处?于是笔者就先从回答什么是操作系统这样的问题入手。

1.2.1 操作系统

什么是操作系统?一般来说,计算机有三个组成部分:硬件、操作系统和应用程序。操作系统是用于直接控制和管理计算机硬件和应用程序资源的一种软件,它存在的目的是为了给用户提供一个更方便高效的环境来使用计算机的硬件和软件资源。因此,操作系统也是一个软件。该软件有两个任务:管理计算机硬件和软件资源以及为用户提供方便高效的使用接口。计算机的硬件资源包括:CPU、内存、外存储器、I/O 设备和高速缓存等设备;软件则既包括用户开发的程序和数据,也包括操作系统本身自带的系统资源、实用程序、库函数和数据库。此外,操作系统还要帮助用户使用这些资源,使得用户更为方便快捷地达到目的。这里所提到的用户包括软件编写人员、系统管理员以及普通终端用户。

在 Unix 操作系统中,用于管理硬件和进程的部分被称为内核。Unix 与用户对话的部分则被称为 shell,即外壳。

1.2.2 初次使用 Unix 操作系统

登录进入系统

在开始使用 Unix 之前,首先要做的事情就是登录(login)。此时,要输入你的用户名和口令以进入系统。

如果你是单用户操作系统的用户,那就只要占有了计算机的硬件资源就可以让它无条件地为你服务,而不存在如何共享资源以及保密的问题。然而,Unix 是一个多用户系统,就必须考虑这些问题。这些工作都是由系统本身的程序来处理的。

当一台安装有 Unix 操作系统的主机启动以后,Unix 的内核就被载入内存,准备等待接受用户的请求。为了识别用户以及完成与用户的通信,内核启动两个程序来达到这一目的: getty 和 login。

一个用户要使用 Unix 操作系统下的资源,首先必须向系统管理员申请账号。系统管理