UNIX 技术丛书

UNIX技术

——系统程序设计篇

刘祖亮 著
新智工作室 改编

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL:http://www.phei.com.cn

UNIX技术丛书

UNIX技术——系统程序设计篇

刘祖亮 著

新智工作室 改编

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

本书由浅入深地介绍了用C或C++进行UNIX系统程序设计的有关内容，并以大量的示例程序相辅，使有一定基础的读者进一步了解UNIX的程序设计。本书前7章主要介绍了UNIX的文件系统，包括：系统调用的基本概念，文件系统的基础知识，与文件系统有关的系统调用，终端机的系统调用，UNIX的快速缓冲区，文件锁定与记录锁定，几种常见的UNIX文件系统。从第8章起，重点转到进程控制系统，介绍了进程的管理，分时与实时系统调度，信号处理，进程间的通讯，UNIX的存储管理系统。最后一章则是对常见问题的讨论与概念澄清。

本书根据台湾和硕科技文化有限公司出版的，由台湾刘祖亮先生编写的繁体版著作《新洞悉UNIX——系统程式设计篇》改编而成，适用于有一定的UNIX系统程序设计经验的人员，也可作为大学计算机系的教材。

图书在版编目(CIP)数据

UNIX技术——系统程序设计篇/刘祖亮著；新智工作室改编. -北京：电子工业出版社，2000.1
ISBN 7-5053-5772-7

Ⅰ.U… Ⅱ.①刘… ②新… Ⅲ.UNIX操作系统-程序设计 Ⅳ.TP316.81

中国版本图书馆CIP数据核字(2000)第01165号

丛 书 名：**UNIX技术丛书**
书　　名：**UNIX技术——系统程序设计篇**
著　　者：刘祖亮
改　　编：新智工作室
责任编辑：李新社
特约编辑：段志钢
排版制作：电子工业出版社计算机排版室监制
印 刷 者：
装 订 者：北京科技印刷厂
出版发行：电子工业出版社　URL：http://www.phei.com.cn
　　北京市海淀区万寿路173信箱　邮编100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：20.25　字数：524.8千字
版　　次：2000年3月第1版　2000年3月第1次印刷
书　　号：ISBN 7-5053-5772-7/TP·2992
印　　数：6000册　　定价：28.00元
版权贸易合同登记号　图字：01-98-0464

改编者的话

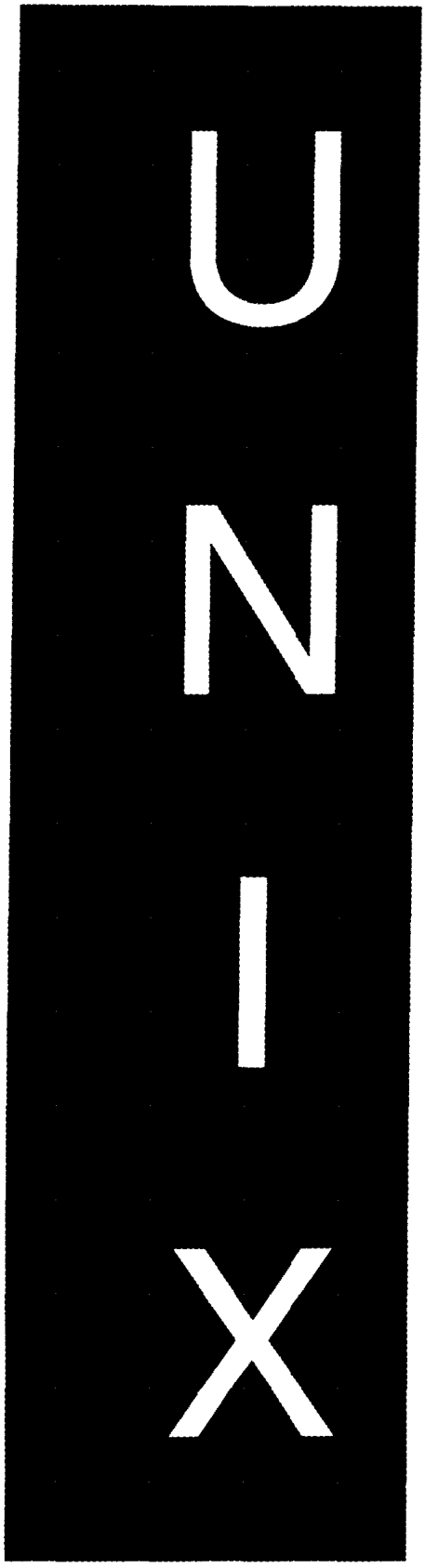
对一些有了 UNIX 系统设计经验，对诸如 pwd、find、ls、chmod、ps、cd 等基本指令已较熟悉的读者，和一些对 UNIX 系统虽然不太熟悉，但学过操作系统课程，又希望在 UNIX 上有所发展的读者，包括计算机系学过数据结构、文件系统等课程的学生，要真正成为水平较高的系统管理员，得有从简单到复杂的过程。一般来说，UNIX 是由两大子系统组成的，即文件子系统和进程控制子系统。为了引到读者逐渐步入“UNIX 程序设计世界”，本书作者以较低层次的程序设计方式出发，即以系统调用方式来讲解有关 UNIX 程序设计方面的知识，然后讲解进程控制子系统。最后为了让读者巩固所学到的 UNIX 知识，加深对 UNIX 系统中一些概念的理解，书中还给出了 UNIX 系统常见问题的解答和讨论。我们相信，本书的这种结构安排，有利于读者掌握 UNIX 的程序设计方法。如果要学习更深层次的程序设计技术可以再参考其他的书籍。

近年崛起的 WindowsNT 采用操作简便和低价格进入服务器操作系统市场，加上其强大的市场宣传攻势获得了部分市场份额。现在，虽然微软的 Windows 操作系统占有很大的市场，但 UNIX 仍然有其发展空间。专家认为 UNIX 的成功来自多方面的因素。UNIX 的主要优势在于技术比较成熟，经实践证明可靠性能高，在伸缩性上比 NT 具有明显优势；而且它仍是目前唯一在各个硬件平台上稳定运行的操作系统。对计算速度要求较高的实时系统，安全性很高和不能间断的银行、电力、航空等领域，以及工业设计、制造设计等特有行业，只有 UNIX 系统才能满足实际需要，这都是众多用户选择 UNIX 的重要原因。随着 Internet 的迅速发展，UNIX 系统平台又显现出其强大的生命力。特别是在网络系统设计中，UNIX 系统有着不可替代的地位。

为了让更多的读者了解 UNIX 系统的程序设计方法，普及 UNIX 系统程序设计知识，我们改编了这本由台湾刘祖亮先生独著，台湾和硕科技文化有限公司出版的图书，此书原书名为繁体版《新洞悉——系统程式设计篇》。在本书的改编过程中，董涛飞先生统一安排了本书的写作计划，并统校了全书。刘文智、路遥、言金刚、段志钢、李强、冯文全、车兰平、官秀梅、涂进等同志对不同章节分别进行了改编。在改编过程中，虽然力求准确和风格一致，但限于学识和水平，谬误之处在所难免，望读者指正。

改 编 者

目　录



0

导论

本章由四个主要的部分构成，第一部分说明本书的写作内容及适用对象。第二部分是全书的大纲，包括了书中各章节的提要及结构。第三个部分则简单地叙述UNIX系统的发展过程，最后一个部分介绍核心程序（kernel）的结构。本书所用到的一些专有名词，也多半在这一章中定义，请读者务必细心阅读。

本书适用对象

本书《UNIX技术——系统程序设计篇》的写作目标是针对下面四种类型的读者：

1. 有使用UNIX系统的经验，对基本指令如 pwd、find、ls、chmod、ps、cd 等已相当熟练，并打算更上一层楼以了解UNIX系统的结构以及内部运作的读者。
2. 虽然对UNIX系统的作业环境并不十分熟悉，可是学习过操作系统的课程，而有心研究UNIX的读者。
3. 有小型电脑系统或大型电脑系统的操作及编程经验，而计划在UNIX系统上"另起炉灶"者。
4. 计算机系二年级以上的学生，学习过数据结构、文件系统等课程。

本书的编排风格

本书的每一章在开始时都有一小段前言，简单介绍全章的要点，读者在进入该章的正文前请先仔细阅读前言部分，以便能迅速掌握重点。

本书写作的目标是尽量务实，因此能以程序表达的概念，我们一定在理论解释后附上说明的示例程序，使读者能了解理论是如何与实际结合的。本书所有的重点文字皆以黑体字表示，指令以斜体字来表现，让您能够一目了然。

阅读本书所必需的背景知识

由于这本书的定位是技术性的读物，因此读者最好有下列背景知识：

1. 具备 C 或 C++ 的编程经验。
2. 对任意一种操作系统十分熟悉，无论是 DOS、OS2 还是 VMS 都可以。
3. 读者最好修过数据结构这门课，要不然至少得知道何谓堆栈、何谓队列以及什么叫作树状结构。

本书的组织结构

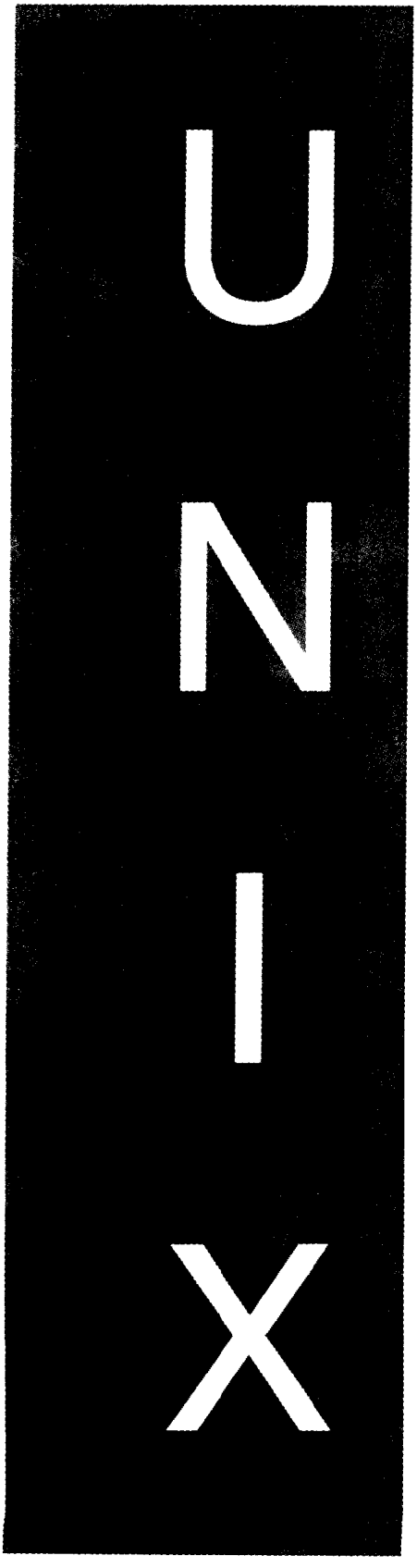
M. J. Bach 在其经典名作《The Design of The UNIX Operating System》中[①]提到UNIX系统是由两大子系统(sub - system)共同组成：**文件子系统(file sub - system)与进程控制子系统(process control sub - system)**；本书也采用此思路来剖析UNIX系统。

由于本书所介绍的程序设计方式属于较低的层次，因此我们会大量地使用系统调用，所以本书的第 1 章就是介绍有关系统调用的基本概念。在入门性的书籍中，作者通常是从用户的角度来看文件系统，而在第 2 章中，则是以操作系统的观点来看文件系统；介绍以文件内在表示方式为主。第 3 章将介绍与文件系统有关的系统调用并辅以完整的程序

① Prentice-Hall 1986, By Maurice J. Bac.

说明。第4章介绍UNIX的终端机输出输入系统、STREAMS以及相关的系统调用。第5章介绍的是UNIX的快速缓冲区。第6章则探讨了文件及记录锁定的奥秘。第7章比较几种常见的UNIX文件系统。

至此,关于文件子系统的说明将暂时告一段落。从第8章起,重点转到进程控制子系统,首先介绍的是进程的定义,并以实际的程序解释fork( )、exec( )、wait( )、exit( )等系统调用的应用。这章的最后介绍了线程(thread)的概念。第9章讨论的主题为分时系统和实时系统的调度。第10章的介绍重点是信号(signal)。第11章及第12章所讨论的主题是进程间的通讯(Inter Process Communication,简称IPC),包括一般管道(pipe)、命名管道(named pipe)、消息队列(message queue)、共享存储器(shared memory)与信号(semaphore)。第13章将介绍UNIX的存储器管理系统。本书的第14章,则是常见问题的讨论与概念的澄清。至于您现在阅读的部分则是对全书所涉范畴的一个完整概要。



1

认识系统调用

我们可以将UNIX系统粗略地分为两个部分:用户层(User Level)与核心层(Kernel Level),请参阅图 1-1。

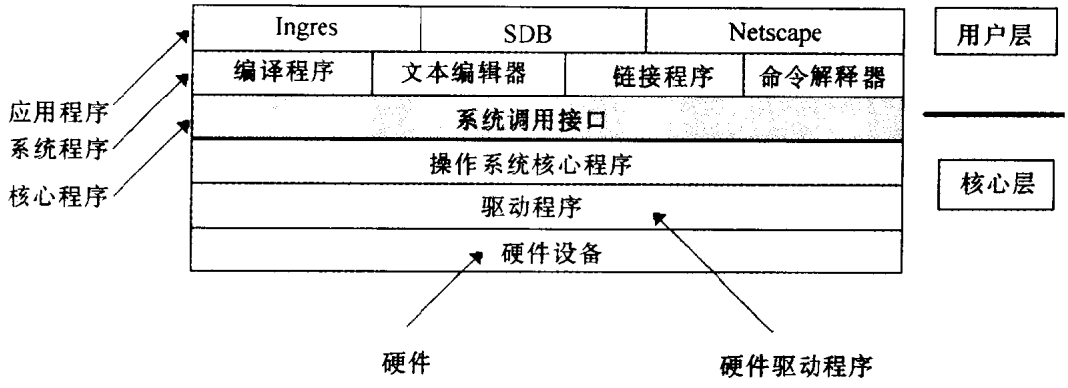


图 1-1 UNIX的用户层和核心层

应用程序所指的范围非常广,它可以是用户使用的任何程序,例如数据库管理系统,也可以是一些套装软件,比方说人事工资管理系统、会计系统等,而UNIX的一些工具程序如 cat、grep 等也都是。系统程序指的是为系统开发提供服务与支持的程序,例如编译程序、文本编辑器,命令解释程序(shell)等都属于此一范畴。

在很多的操作系统(特别是早期开发的)中,其命令解释器(Command Interpreter,在UNIX系统就是 shell,在 DOS 就是 command.com)往往是系统核心的一部分,而UNIX则不然,它的命令解释器 csh 或 sh 都是一个独立运行的程序,所以把它与系统程序归到同一类。

核心层就是UNIX的核心程序。所有的进程管理、存储器管理、文件系统管理及设备驱动程序(Device Driver)等皆属于核心层范围。核心程序直接与电脑硬件沟通,而输出输入则是外围驱动程序的工作,例如主控台驱动程序负责接收从键盘输入的字符,并将要显示的字符输出到终端机的显示器屏幕上,硬盘驱动程序则负责进行硬盘输出/输入的工作,例如移动磁头等。这样的系统结构会促使因硬件不同而不一样的部分多半集中在外围驱动程序部分,所以在进行系统移植(porting)时,可以省掉全面改写的麻烦。再者,整个核心程序有将近九成是以 C 语言编写的,仅有一小部分是混合语言编写的(主要是系统启动时与硬件设备关系密切的部分),高级语言应用在操作系统设计上使得系统移植的花费降到最低;使得UNIX成了最容易移植的多用户多任务系统。

何谓系统调用

在图 1-1 中,您会发现在用户层与核心层之间有一个中间地带,称之系统调用(system calls),它是用户层与核心层之间的接口。**系统调用是一群预先定义好的模块(多半由混合语言所编写),它们提供一条管道让应用程序或一般用户能由此得到核心程序的服务**,例如外围设备的使用、程序的执行、文件的传输等。换言之,系统调用担任一个保护人兼中介者的角色。它一方面在核心程序与用户之间传递信息,另一方面也可防止因用户的程序不慎而破坏核心程序内的一些子程序(routine)。

由于UNIX是多用户多任务系统,因此能够在同一时间内为一个以上的用户服务,也可以同时执行好几个进程,就算只有一个用户在使用,也能够同时执行数个程序;而且系

统本身还有一些进行数据维护的程序会定期执行(例如 crontab,sync 等),因此UNIX的系统资源可由许多个进程所共享,不像 DOS 程序,它是单用户单任务系统,所有资源全由单一的用户独享,使得程序员可以任意改变或读取系统的状态,也可以通过对 BIOS 的直接调用向外围设备下达输出输入的指令。在多用户多任务的系统里,这种做法会破坏整个核心程序的运作秩序(您想想,当您的报表打印到一半时,忽然有人调用 BIOS 来打印他的数据,这时的报表输出将会是什么样的!),为了避免这种情形的出现,在UNIX下,程序员必须通过系统调用来取得核心程序的服务,让所有向系统要求的服务,无论是软件还是硬件,一律得通过核心程序本身来完成,如此可使一切的行动都在核心程序监控之下。

系统调用的使用

在UNIX系统中,系统调用多半由 C 语言所提供的函数库来完成,当然使用混合语言也能完成同样的工作,只是一般而言不是那么普遍。由于在 C 语言里使用系统调用的形式与调用一般的函数几乎没有什么区别,因此有经验的程序设计人员使用起来会觉得驾轻就熟。

根据系统调用的功能与性质,笔者把系统调用分为六类:

1. 进程的管理
2. 存储器的管理
3. 文件系统的管理
4. 输出与输入设备的管理
5. 进程间的通讯(IPC)
6. 系统信息的获取

对于大部分的读者,笔者建议不妨将系统调用子程序当成 C 语言的函数看待,这样在学习上会觉得方便些。

系统调用的执行

图 1-2 描述了执行系统调用的整个过程,在探讨本图的意义之前,先一起来看几个重要的专有名词。

1. 进程表(process table):在核心程序内有一个进程表,每个执行中的进程在这个表单里都设有一个栏,这个栏用来记载各进程的执行状态。
2. 用户区(user - area 简称 U - area):每个进程在启动后,核心程序都会为这个新进程分配一块存储空间。该空间用来记录有关此进程的信息,这些信息在进程执行时,可供核心程序使用。这些信息包括:
   - 这个进程打开了哪些文件。
   - 根目录为何,当前目录为何。
   - 本子系统调用有几个参数。
   - 进程的堆栈段、数据段、及程序段的大小。
   - 一个指向进程表的指针;通过这个指针所指向的地址,核心程序可以找到进程