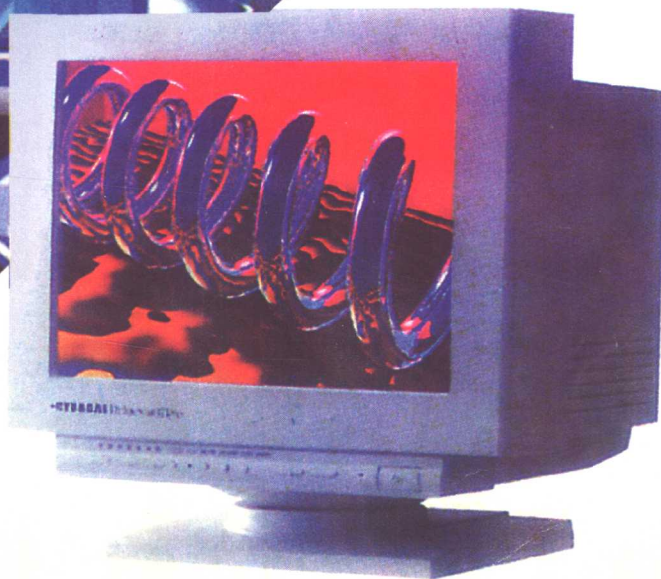


CAO ZUO XI TONG FEN XI YU SHE JI

操作系统分析 与设计

李彤 沈勤祖 王黎霞 编著



云南大学出版社

操作系统分析与设计

李 彤 沈勤祖 王黎霞 编著

云南大学出版社

责任编辑:周永坤

封面设计:丁群亚

内 容 简 介

本书以 Unix System V 为对象,全面系统地分析了 Unix System V 的主要数据结构 and 算法,介绍了操作系统的模块组合结构、进程分层结构、层次管程结构和面向对象的结构及相应的设计方法。通过对 Unix System V 的分析,力求通过具体的材料来体现操作系统的基本原理和设计思想。第一章是全书的总纲;第二章至第六章分别介绍了 Unix System V 的文件系统、进程管理、存储管理、设备管理和进程通讯的实现原理、内部数据结构和算法;第七章介绍了 Unix 的二类用户接口;第八章简要介绍了操作系统的结构及设计方法。每章后配有大量的习题。

本书可以作为计算机类专业及相关专业研究生和本科生的教材和教学参考书,也可供广大计算机工作者阅读和参考。

书 名: 操作系统分析与设计

编 著: 沈勤祖 王黎霞

标准书号: ISBN 7-81025-646-7/TP·23

出 版 者: 云南大学出版社出版发行

地 址: 云南大学校内

邮政编码: 650091

排 印 者: 云南大学出版社印刷厂

云南大学出版社印刷厂印刷

版本记录: 787×1092 16 开本 15.9 印张 406.4 千字

1996 年 12 月第一版 1996 年 12 月第一次印刷

印数: 0—5 000 册

定 价: 19.00 元

前 言

操作系统是现代计算机系统中最重要、最核心的系统软件之一,它在整个计算机系统的软件体系结构中占有中心地位。因此,不论是科研机构、高等学校、计算机厂商,还是普通用户,都十分重视对操作系统的研究、分析、开发和应用,都十分重视充分发挥操作系统的作用。作为计算机系统资源的管理者,操作系统不但要对计算机系统中的各种资源进行有效的管理,充分发挥各种资源的使用效率,而且还要提供一个方便地使用计算机系统的良好用户界面。早期的操作系统由于软硬件技术发展的限制,采用汇编语言来编写,追求的是系统的功能和效率。随着软硬件技术的发展,人们日益认识到了操作系统结构的重要性,同时技术进步也为人们使用高级语言来编写操作系统提供了物质基础。

Unix 操作系统是 AT&T 公司 Bell 实验室的 Ken Thompson 和 Dennis Ritchie 于 1969~1970 年间研制成功的一个分时操作系统。其后 Dennis Ritchie 研制成功了 C 语言,并将 Unix 用 C 语言重新改写加以实现。在此后的 20 多年间,Unix 不断发展完善,目前它几乎已运行于从巨型机到微型机的各种硬件平台,成为多用户系统事实上的工业标准,被公认为开放式系统结构的核心。这一切成就在很大程度上得益于用 C 语言来编写 Unix。二者互相促进,相得益彰。可以说,没有 C 语言,就没有 Unix 的今天;不用来书写 Unix,也不能把 C 语言的功能与优良性质发挥得如此淋漓尽致。作为 Unix 的研究者和学习者,我们常常为其精美的设计而拍案叫绝,也常常从 Unix 那里领略到了在其它地方所领略不到的 C 语言的丰采。我们应该感谢 Dennis Ritchie 和 Ken Thompson,正是由于他们的杰出贡献,不但为人类提供了一个优秀的操作系统,而且使 Unix 的影响远远超出了操作系统的范畴,对计算机科学产生了巨大而深远的影响。

对操作系统的分析与设计是研究、开发操作系统过程中的二个互相联系又互不相同的方面。分析是被动的,而设计是主动的。本书以 Unix System V 操作系统为对象,全面系统地分析了 Unix System V 的主要数据结构和算法。通过这些分析,读者可以领会到 Unix 的设计原理。但本书不是 Unix 的说明书,材料的取舍服从全书目标的需要。在最后一章还介绍了操作系统的四种体系结构及其设计方法。

本书是为研究生和本科生编写的一本教材。其前身作为讲稿曾在不同层次的人员中讲授,受到了普遍的欢迎。本书是按 80 个学时编写的,若按 60 学时讲授,建议免讲部分内容。

作为教材,在材料组织上,我们力求通过具体的材料来体现操作系统的基本原理,通过对具体操作系统的分析来体现其设计思想。第一章是全书的一个总纲。第二章到第六章是对 Unix 内核的分析。第七章介绍了 Unix 的二类用户接口:系统调用和 Shell 命令接口。第八章介绍了操作系统的模块组合结构、进程分层结构、层次管程结构和面向对象的结构及其设计方法。在介绍 Shell 命令时,由于本书的目的不是要教会读者具体使用 Unix 系统,而是在于从用户接口的角度来展示 Shell 命令的一些特征,因此仅作了简单扼要的介绍。

本书第一章至第六章由李彤同志编写,第八章由沈勤祖同志编写,第七章由王黎霞同志编写,最后由李彤同志进行了统稿。

读者在阅读本书时,应当具有操作系统原理方面的知识和使用 Unix 方面的知识。若不具备使用 Unix 方面的知识,建议首先阅读 § 7.3 和 § 7.4 节。

本书在编写过程中得到了云南大学计算机科学系田志良教授、刘惟一教授、周行仁教授,中国科学院软件研究所仲萃豪研究员的关心、鼓励 and 真诚的帮助,得到了云南大学出版社龙宝珍老师的大力支持,本书的责任编辑周永坤老师做了大量艰苦细致的工作,在此作者表示衷心的感谢。此外,我们还得到了云南大学计算机科学系谢戈高级实验师、图书馆李汉斌老师、92 级本科生张晶同学的真诚帮助,在此一并致谢。

由于水平所限,书中肯定存在不少缺点错误,作者真诚希望广大读者、专家、同行批评指正。

编 著 者

1996 年 10 月于云南大学

目 录

第一章 绪论	(1)
§ 1.1 操作系统概述	(1)
§ 1.2 Unix 综述	(2)
1.2.1 Unix 发展概况	(2)
1.2.2 Unix 特点	(3)
1.2.3 Unix 分层体系结构	(5)
§ 1.3 Unix 内核概述	(5)
1.3.1 Unix 内核结构	(5)
1.3.2 Unix 内核功能简述	(6)
习题一	(7)
第二章 Unix 文件系统	(8)
§ 2.1 概述	(8)
§ 2.2 文件系统的数据结构	(9)
2.2.1 文件系统的存储结构	(9)
2.2.2 i 节点	(10)
2.2.3 目录文件	(12)
2.2.4 超级块	(13)
2.2.5 用户打开文件表	(13)
2.2.6 系统打开文件表	(14)
2.2.7 安装表	(15)
2.2.8 各数据结构间的关系	(15)
§ 2.3 文件系统存储资源管理	(15)
2.3.1 空闲块的管理	(16)
2.3.2 磁盘 i 节点的管理	(21)
2.3.3 内存 i 节点的管理	(26)
2.3.4 文件表的管理	(28)
§ 2.4 正规文件的物理结构	(29)
2.4.1 地址索引表	(30)
2.4.2 地址映射	(30)
§ 2.5 目录文件	(34)
§ 2.6 与文件有关的系统调用	(36)

2.6.1 概述	(36)
2.6.2 系统调用 open, creat 和 close	(37)
2.6.3 系统调用 read 和 write	(42)
2.6.4 系统调用 link 和 unlink	(45)
2.6.5 系统调用 mount 和 umount	(47)
2.6.6 系统调用 chdir 和 chroot	(52)
2.6.7 系统调用 mknod	(54)
2.6.8 系统调用 fstat 和 stat	(55)
2.6.9 系统调用 dup	(56)
2.6.10 系统调用 lseek	(57)
2.6.11 系统调用 pipe	(58)
2.6.12 系统调用 chmod 和 chown	(59)
§ 2.7 与文件有关的系统调用实例	(61)
习题二	(65)

第三章 Unix 进程管理

§ 3.1 进程结构	(68)
3.1.1 进程控制块 PCB	(68)
3.1.2 进程的上下文	(73)
3.1.3 进程的状态及 转换	(73)
3.1.4 数据结构间的关系	(76)
3.1.5 系统用变量	(76)
§ 3.2 进程上下文的保存与切换	(79)
3.2.1 中断和异常	(79)
3.2.2 系统调用接口	(79)
3.2.3 上下文切换	(80)
§ 3.3 进程控制	(81)
3.3.1 进程树	(81)
3.3.2 进程的创建	(82)
3.3.3 执行一个文件	(84)
3.3.4 进程的睡眠与唤醒	(85)
3.3.5 进程的终止	(89)
3.3.6 等待子进程终止	(91)
§ 3.4 进程调度	(91)
3.4.1 调度原理	(92)
3.4.2 调度的实现	(93)
§ 3.5 与进程有关的系统调用及实例	(95)
3.5.1 与进程属性有关的系统调用	(95)
3.5.2 与时间有关的系统调用	(97)
3.5.3 系统调用实例	(100)

习题三	(105)
第四章 Unix 存储管理	(107)
§ 4.1 进程的虚拟地址空间	(107)
4.1.1 区	(107)
4.1.2 页和页表	(108)
4.1.3 对进程虚拟地址空间的操作	(109)
§ 4.2 交换	(115)
4.2.1 交换空间的管理	(115)
4.2.2 进程的换出	(118)
4.2.3 进程的换入	(120)
4.2.4 进程交换的实现	(120)
§ 4.3 进程正文段的共享	(124)
4.3.1 数据结构	(124)
4.3.2 算法	(125)
§ 4.4 请求调页	(126)
4.4.1 基本原理	(126)
4.4.2 数据结构	(127)
4.4.3 动态地址变换	(129)
4.4.4 页故障	(129)
4.4.5 换页进程	(133)
§ 4.5 交换策略与请求调页策略的结合	(135)
§ 4.6 与存储管理有关的系统调用及实例	(135)
习题四	(137)
第五章 Unix 设备管理	(140)
§ 5.1 概述	(140)
§ 5.2 中断和异常	(140)
5.2.1 中断的分类	(141)
5.2.2 异常的分类	(141)
5.2.3 中断和异常的处理	(142)
§ 5.3 缓冲区管理	(147)
5.3.1 缓冲池结构	(147)
5.3.2 缓冲区的分配与释放	(150)
5.3.3 块的读写	(152)
§ 5.4 设备驱动程序的接口	(155)
5.4.1 设备开关表	(156)
5.4.2 设备驱动程序的接口	(157)
5.4.3 系统调用与驱动程序的接口	(157)
5.4.4 中断处理程序	(161)

§ 5.5 字符设备驱动	(161)
5.5.1 数据结构	(161)
5.5.2 对字符缓冲区队列的操作	(162)
5.5.3 终端设备驱动	(169)
§ 5.6 块设备驱动	(177)
§ 5.7 与设备有关的系统调用实例	(179)
习题五	(180)
 第六章 Unix 进程通讯	(182)
§ 6.1 软中断	(182)
6.1.1 软中断检测与处理的时机	(182)
6.1.2 软中断的种类	(182)
6.1.3 软中断的检测与处理	(184)
6.1.4 与软中断有关的系统调用及实例	(188)
§ 6.2 管道	(190)
6.2.1 无名管道	(190)
6.2.2 有名管道	(193)
§ 6.3 进程间通讯 IPC	(195)
6.3.1 消息机制	(195)
6.3.2 共享存储区机制	(202)
6.3.3 信号量机制	(206)
习题六	(212)
 第七章 Unix 用户接口	(214)
§ 7.1 系统调用	(214)
§ 7.2 Shell 进程	(218)
7.2.1 Shell 概述	(218)
7.2.2 Shell 流程	(219)
7.2.3 Shell 进程的生成	(221)
§ 7.3 Shell 常用命令	(221)
7.3.1 与用户有关的命令	(222)
7.3.2 与文件系统有关的命令	(223)
7.3.3 进程与通讯命令	(226)
7.3.4 后台命令	(227)
7.3.5 管道与重定向命令	(228)
§ 7.4 Shell 程序语言	(228)
习题七	(231)
 第八章 操作系统的设计	(232)
§ 8.1 操作系统设计概述	(232)

§ 8.2 结构设计思想的产生	(233)
§ 8.3 操作系统的模块组合结构	(234)
§ 8.4 操作系统的进程分层结构	(236)
8.4.1 进程分层结构的产生	(236)
8.4.2 层次划分的依据	(237)
8.4.3 进程分层结构的优缺点	(237)
§ 8.5 操作系统的层次管程结构	(238)
8.5.1 概述	(238)
8.5.2 层次管程结构的优缺点	(239)
§ 8.6 面向对象的结构设计	(239)
8.6.1 面向对象的基本概念	(239)
8.6.2 面向对象操作系统的结构设计	(241)
习题八	(242)
参考文献	(243)

第一章 绪 论

操作系统是计算机系统中最重要、最核心的软件系统之一。它对计算机系统内的各种软、硬件资源进行有效的管理和控制,并提供友好的用户界面。现代计算机系统中,不论是巨型机、大型机、中小型机还是微型机,都毫不例外地配置有一种或多种操作系统。编译程序、数据库管理系统及各种应用系统都必须有操作系统的支持才能运行。因此,对操作系统结构的分析对于深刻理解操作系统的原理、掌握现有操作系统、设计新的操作系统、研究新的设计方法以及对其它系统软件的分析与设计均具有重大的意义。

§ 1.1 操作系统概述

操作系统出现于 50 年代末期,至今已有近 40 年的历史。最早出现的操作系统的雏形称为监控程序(Monitor)和执行系统(Executive System)。随着软硬件技术的发展,操作系统也逐渐走向成熟。60 年代中期以来,许多优秀的操作系统相继问世,比较有代表性的有 MVS、Unix、CTSS、VM/SP、OS/360、OS/2、VMS、DOS 等。

操作系统的职能是对计算机系统中所有软硬件资源进行有效的管理和控制,合理地组织计算机系统的工作流程,充分发挥各种资源的使用效率,并提供用户友好的用户界面。因此,从资源管理和用户接口的角度出发,操作系统具有以下五个基本功能:

- (1)处理机管理;
- (2)存储管理;
- (3)设备管理;
- (4)文件系统管理;
- (5)用户接口。

在操作系统近 40 年的发展历程中,为满足不同的应用要求以及兼顾宿主主机系统硬件的特点,再加之历史的背景因素,形成了各种不同类型的操作系统。根据使用环境和对作业处理方式的不同,可将操作系统分为以下六种基本类型:

- (1)单用户操作系统;
- (2)批处理操作系统;
- (3)分时操作系统;
- (4)实时操作系统;
- (5)网络操作系统;
- (6)分布式操作系统。

随着硬件技术的发展,操作系统也不断改进,功能日益完善,出现了通用操作系统,即兼有多道批处理、实时、分时功能中二种以上功能的操作系统。例如 MVS 就是一个以批处理为主,但也同时支持分时使用的通用型操作系统。

进入 90 年代,计算机硬件技术更以前所未有的速度迅猛发展,各种软件系统规模也越来越庞大,这更要求操作系统提供更强有力的支持。操作系统正朝着分布式、智能化、个人机化方

向发展,对用户界面的要求也越来越高。可以预计,随着多媒体技术的发展和信息高速公路的建成,必将对操作系统提出更高的要求。

同一切大型软件系统一样,操作系统的研制过程要经历系统分析、设计、实现、测试和维护等若干阶段。由于设计方法的不同,导致操作系统内部结构也不一样。操作系统的结构大致可分为以下四类:

- (1)模块组合结构;
- (2)进程分层结构;
- (3)层次管程结构;
- (4)面向对象结构。

操作系统是具有并发、共享、虚拟和不确定性等四个基本特征的大型软件系统,模块间的接口相当复杂,信息交换十分频繁,对可靠性有很高的要求。因此,对操作系统结构和结构设计方法的研究就显得格外重要,受到了软件工程界的普遍重视。

分析现有操作系统的结构是研究操作系统结构的重要方面,是对操作系统进行结构设计的基础。对操作系统结构的分析不仅能使读者对分析对象有深入透彻的了解,而且对于分析和设计其它系统软件也具有重大的借鉴意义。

Unix 操作系统是一个多用户分时操作系统。由于它的绝大部分代码是用 C 语言编制的,故可移植性极好。目前它已运行于从巨型机到微型机的各种主要硬件平台,成为多用户系统和 workstation 系统事实上的工业标准。Unix 核心短小精干,功能却十分强大,并充分体现了操作系统的基本原理,具有操作系统的各项基本功能,十分适合于作为分析对象。故本书采用目前最新的 Unix System V 作为分析对象,全面系统地分析了 Unix System V 的内核结构,对进程管理、存储管理、文件管理、设备管理等子系统的重要数据结构和算法进行了深入的剖析,介绍了 Shell 的实现原理及 Unix 用户接口。通过对 Unix 的分析,读者也可以深入了解并深刻领会到 Unix 的设计思想和设计过程。在对 Unix 深入认识的基础上,本书简要介绍了操作系统的几种重要结构及对应的结构设计方法。

§ 1.2 Unix 综述

1.2.1 Unix 发展概况

60 年代,由于计算机硬件技术为软件的发展提供了广阔的舞台,计算机界掀起了一股开发大型软件的热潮,表现在操作系统研制上的一个例子就是 1965 年贝尔实验室、通用电气公司和麻省理工学院合作研制称为 Multics 的新操作系统。这个系统功能庞大,追求大而全,开发人员众多,要求提供支持多个用户同时进行数据处理的能力。其名字前缀 Multi 即表示多个的意思。由于种种原因, Multics 的目标未能完全达到,贝尔实验室退出了该研制项目。贝尔实验室中许多参加 Multics 项目的人员后来参与了 Unix 系统的研制。

贝尔实验室退出 Multics 项目后,原来参加该项目的 Ken Thompson 和 Dennis Ritchie 等人设想设计一个简明的系统,并将在 Multics 的分时终端上编写的“Space Travel”游戏程序移置到 PDP-7 上。移植时遇到了许多困难,这促使 Thompson 等人动手编写自己的操作系统,并于 1969 年开始在 PDP-7 上开发这个系统,于 1970 年投入运行。这个系统开发人员少,功

能清晰,追求小而精,被开发小组的另一个成员 Brian Kernighan 戏称为 Unix。Unix 是相对于 Multics 的双关语,前缀 uni 表示“一个”、“单独”的意思,而 multi 则表示“多个”的意思,Unix 中的 X 与 Multics 中的 cs 发音又相同。于是 Unix 诞生了。

Unix 得到实用是在 1971 年,运行环境是 DEC 公司的 PDP-11 计算机,它满足了贝尔实验室的正文处理要求。此时的 Unix 系统是用汇编语言写的。

1973 年,Ritchie 将 Unix 系统用 C 语言重写了一遍,从而使 Unix 成为第一个从汇编语言表达转向用高级语言表达的操作系统,为 Unix 的普及起到了极为关键的作用。C 语言是 Ritchie 对 Thompson 的解释性 B 语言进行改进而提出来的。它允许定义数据结构、说明数据类型,表达能力很强,同时又保留了一些较低级语言的特性,容易编译成高效的目标代码,很适合于编写系统软件。此后 C 语言和 Unix 相互依存,相互促进,共同成为了著名的软件系统。同年,Thompson 和 Ritchie 合写的论文“The Unix Time-Sharing System”提交美国计算机协会(ACM)的第四次操作系统原理讨论会。经修改后,该文发表在 1974 年 7 月的《Communication of ACM》杂志上。至此,Unix 正式向外公布。

1977 年,Unix 被移植到非 PDP 硬件 Inter Data 8/32 计算机上。很快其它公司也开始把 Unix 移植到其它机器上,并对 Unix 作了许多改进。用 C 语言书写 Unix 的好处得到了验证。于是 Unix 演变出许多变种来,即使在 AT&T(贝尔实验室所在公司)内部也产生了几种变种。为统一版本,贝尔实验室把它们结合到一个标准系统中,这就是 Unix System III。

1980 年,贝尔实验室公布了为 VAX 11/780 计算机编写的 Unix 32V。同年,在 Unix 32V 的基础上,美国加州大学伯克利分校发表了 VAX 11 计算机专用的 Unix 类操作系统 BSD4.0 和 BSD4.1。从此,Unix 分裂为二大流派,即以 AT&T 为代表的 Unix 和以加州大学伯克利分校为代表的 BSD 二大流派。

1983 年,AT&T 公布了 Unix System V,这就是今天所流行的 Unix System V 的最早版本。同年,Thompson 和 Ritchie 因 Unix 而获 ACM 图灵奖。

1985 年,Microsoft 公司将 Unix 改造为适用于 16 位 PC 机的变种 Unix 系统 Xenix。Xenix 曾经为 Unix 类系统在微机领域中的普及作出了巨大贡献。

此后几年,各种 Unix 类系统如雨后春笋般出现,使 Unix 已运行于从巨型机到微型机的各种主要硬件平台。正是在这种背景下,美国 IEEE 组织成立了 POSIX 委员会,专门从事 Unix 的标准化工作。此前,UI 和 OSF 两大集团已经开始了 Unix 的标准化工作,它们定义出了什么是 Unix 系统的统一标准,即一个可以运行 Unix 应用软件的操作系统就是 Unix。从而,Unix 系统的统一关键就在于提供一个标准用户界面,而不在于内部如何实现。

1989 年,SCO 公司推出了可在基于 80X86 芯片的微机 Unix System V 操作系统。由于微机性能大幅度提高,价格却不断下降,致使微机正迅速普及,走向千家万户,从而使 SCO Unix System V 为 Unix 的普及发挥了重要作用。

1.2.2 Unix 特点

Unix 获得巨大成功,同它具有的优良特性是分不开的。Unix 的主要特点表现在以下几个方面:

1. 短小精悍的内核,强大的系统功能

Unix 的设计者认为操作系统的内核应当只完成最基本的功能,应该对外提供机制而不是具体的服务。因此,他们对现有的实现技术作了恰如其分的选择和精巧的实现,具体服务由大

量的核外实用程序完成。

由于内核短小简洁,因此只需占用很小的存储空间,故有条件常驻内存,从而保证了整个系统的高效运行。而核外程序充分利用内核提供的支持,向用户提供各种良好的服务。它们只在运行时才进入内存,因而可以开发出大量的核外程序为用户服务,而不必顾虑内存是否承受得住。内核和核外程序有机结合起来成为一个整体,其功能可以同大型系统相媲美。

2. 良好的可移植性

大多数操作系统是用汇编语言编制的。由于汇编语言十分依赖于硬件环境,致使这些操作系统很难适用于不同的硬件平台。而 Unix 绝大多数代码是用 C 语言编制的,因而很容易在不同硬件平台之间移植。这一特点为 Unix 系统的普及发挥了重要作用。

3. 树形结构的文件系统

Unix 采用树形目录的文件系统结构管理文件,层次清晰,有效地解决了文件重名问题,搜索速度快,具有良好的安全性、保密性和可维护性。它提供了文件系统的装卸功能,使用户可以同时使用多个文件系统,极大地提高了文件系统的灵活性。

4. 统一看待外设和文件

在 Unix 中,文件和外设都被统一看成文件,它们在用户面前具有相同的操作方式,使用相同的保护机制,都是无结构的字符流。这样做既简化了系统设计又便于用户使用。

5. I/O 重定向和管道

通过 I/O 重定向,可以指定命令或程序从何处得到输入和将结果送到何处去,从而改变了缺省的输入源位置和输出目标位置。管道让一个命令或程序的输出成为另一个命令或程序的输入。这两项功能极大地提高了用户使用 Unix 的灵活性。

6. 良好的用户界面

Unix 提供 Shell 和系统调用二种用户界面。前者是一个命令解释器,它将用户输入的命令传递给内核,又将内核的执行结果转达给用户。后者是 Unix 提供给程序员在源程序一级使用系统功能的手段。另外,前者还是一个命令语言,它提供了条件、循环等控制语句,用户使用它能够高效率地在系统上进行工作。采用后台方式还可向系统提交类似批处理方式的作业。

7. 开放式的操作系统

Unix 允许用户自己编写工具和实用程序并以非常自然的方式加入系统中;提供有丰富的通信和网络软件,提供有 TCP/IP 协议来实现同其它网络的互连。Unix 倡导的开放性对整个计算机领域产生了深远的影响。

8. 强有力的软件开发环境

Unix 除了具有操作系统的基本功能外,还提供有大量的软件开发工具,如源代码控制系统(SCCS)、词法分析器自动生成器(Lex)、优化 C 编译器和源代码调试工具等。为用户进行软件开发提供了强有力的支持。

9. 丰富的系统应用软件

由于 Unix 的良好特性,它获得了广大软件厂商的支持。目前运行于 Unix 系统之上的系统应用软件覆盖范围极广,有电子表格、数据库管理系统(DBMS)、计算机辅助设计(CAD)、计算机辅助制造(CAM)和计算机辅助软件工程(CASE)等,为用户的应用开发提供广泛的支持工具。

10. 丰富的实用程序

Unix 提供了丰富的实用程序来为用户服务。这些实用程序使用非常灵活,将它们组合起

来可以完成特殊的功能。例如文本编辑器(vi、ed 等)、文本格式化工具(nroff、troff)等。

1.2.3 Unix 分层体系结构

Unix 系统可分成三层。最内层是 Unix 系统的核心,它提供进程管理、存储管理、设备管理和文件管理等各项功能。中间一层是 Shell 和实用程序层,它在内核和用户程序之间起到中间纽带作用。并提供各种实用程序和库函数服务。最外层是应用层,它由用户程序构成。Unix 系统建立在具体机器硬件的基础之上,构造了分层的虚拟机,为用户提供独立于具体机器硬件的各种服务。Unix 分层体系结构如图 1.1 所示。

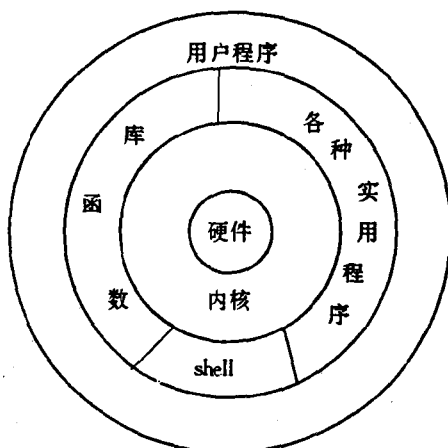


图 1.1 Unix 分层体系结构

§ 1.3 Unix 内核概述

Unix 内核是 Unix 系统的核心部分,它是机器硬件的第一层扩展,构成了第一层虚拟机。Unix 内核完成进程管理、存储管理、设备管理和文件管理等功能,对外提供称为系统调用的服务。系统调用是一般用户取得内核服务的唯一手段。

1.3.1 Unix 内核结构

从系统构成的角度我们可将 Unix 系统大致看成三级:用户级、核心级、硬件级。用户级包括 Shell、各种实用程序、标准库函数等。核心级即 Unix 内核,它主要完成进程、存储、文件和设备的管理工作。硬件级即 Unix 所安装的宿主计算机硬件系统。Unix 内核的结构如图 1.2 所示。

Unix 内核采用以全局变量为中心的模块结构,因而系统结构较为复杂,模块间循环调用和文件之间共享全局变量等现象比较突出。因此图 1.2 中各模块的划分是逻辑上的,而在实际实现时,它们之间互相交叉调用,模块的划分非常模糊。

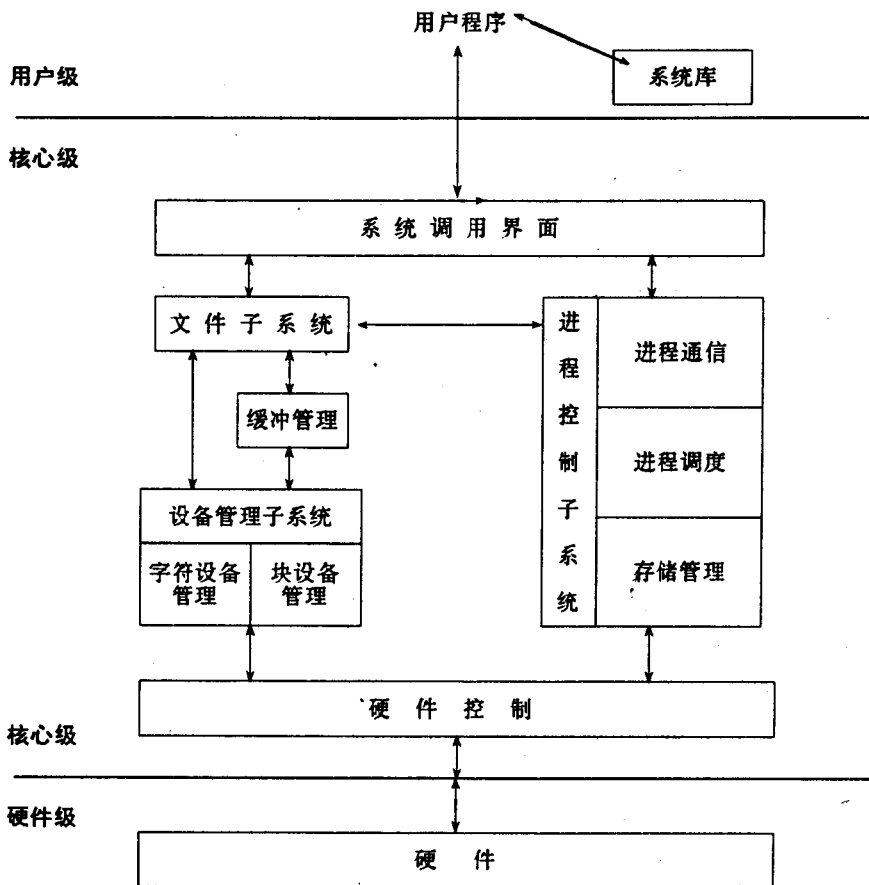


图 1.2 Unix 内核结构

1.3.2 Unix 内核功能简述

一、文件子系统

Unix 文件分为四类：正规文件、目录文件、设备文件和管道文件。正规文件是存放程序、数据等的文件。从用户角度看，它是一个无格式的字符流文件。用户按照自己的格式来解释字符流，获得自己所需的信息。目录文件是存放文件系统中各个目录信息的文件，它也是字符流式文件，系统将其解释为文件目录，所有的目录文件构成整个 Unix 文件系统的树状结构。设备文件代表一个物理设备，用户可以按处理正规文件的方式对其进行处理，但设备文件除了有关文件管理的信息外，并不占据实际的物理存储块。管道文件用来存放管道数据。

文件子系统负责管理文件。具体功能有：

- (1) 空闲文件存储空间的管理；
- (2) 为文件分配文件存储空间；
- (3) 回收文件释放的文件存储空间；
- (4) 文件存取控制；
- (5) 搜索文件；
- (6) 为用户提供系统调用服务。

二、进程控制子系统

进程是一个具有独立功能的程序对其所处理的数据在处理机上的执行过程。在 Unix 中,当用户在 Shell 状态下键入一个可执行的文件(程序)名时,该文件(程序)即以进程的形式在 Unix 系统中运行,完成用户要求的任务。

进程控制子系统负责进程的管理。具体功能为:

- (1)进程的创建;
- (2)进程的调度;
- (3)进程间的通信;
- (4)进程间的同步控制。

三、设备管理子系统

设备管理子系统完成进程和外设间数据交换的功能。在 Unix 中,外设被看成是文件(设备文件),而且文件信息本身也是存放在外设上的。因此,当进程访问文件时,必然发生外设与内存之间的数据流动。设备管理子系统中存放有许多设备驱动程序,它们驱动外设完成真正的输入输出工作。

四、存储管理子系统

存储管理子系统主要负责内存的管理。它对一般用户来讲是看不到的。其主要功能有:

- (1)管理内存的空闲空间;
- (2)对交换区空间(一般在磁盘上)进行管理;
- (3)对虚拟存储空间进行管理。

物理内存空间是十分有限的。为了充分提高宝贵的内存的使用效率,Unix 后期版本只把进程的一小部分程序和数据放在内存中,而把剩余的程序和数据放在外存,然后采用交换和请求调页的存储管理策略实现对内存的管理,提供用户比物理内存大得多的虚拟地址空间。Unix 的早期版本仅采用交换技术进行内存管理。

本书是作为教材来编写的,其目的在于通过对一个实际的操作系统分析让读者了解到操作系统的结构和设计方法。因此,在材料取舍上,力求做到服务于这个目标,而不过于追求微小的细节。读者在阅读本书时,要同操作系统原理结合起来,充分理解本书所体现的具体操作系统的设计原理,达到综合贯通。

习 题 一

- 1.1 什么是操作系统? 操作系统有哪些基本功能?
- 1.2 操作系统的结构大致可分为哪几类? Unix 的结构有什么特点?
- 1.3 Unix 具有哪些主要特点? 使 Unix 在全世界广泛流行的根本原因是什么?
- 1.4 Unix 系统可以分为哪几层? 各层的主要功能各是什么?
- 1.5 什么是 Unix 内核? Shell 属于 Unix 内核吗?
- 1.6 Unix 内核的功能主要包括哪些方面? 每一个方面各有哪些具体内容?
- 1.7 通过阅读本书,你希望达到什么目标? 本书所要达到的目标是什么? 本书的目标与你的目标之间有何差异? 你将采取什么措施来弥合二个目标之间的差异?