

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

操作系统 教程

A Course of Operating Systems

孟静 唐志敏 编著

- 每个知识点3W都真正讲清讲透
- 真正让OS工作过程在脑中转起来
- 独特的“硬件相关/应用无关”定义
- 真正做到理论处处密切联系实际



名家系列

 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

操作系统 教程

A Course of Operating Systems

孟静 唐志敏 编著



名家系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

操作系统教程 / 孟静, 唐志敏编著. —北京: 人民邮电出版社, 2009. 4
21世纪高等学校计算机规划教材
ISBN 978-7-115-19524-1

I. 操… II. ①孟…②唐… III. 操作系统—高等学校—教材 IV. TP316

中国版本图书馆CIP数据核字 (2009) 第017137号

内 容 提 要

本书全面系统地介绍操作系统通用原理内容, 并采用 Linux 作为贯穿全书的完整实例。全书共 7 章, 依次为操作系统概论、处理机管理、内存管理、外存管理和文件系统、设备管理、进程通信、操作系统高级教程等内容。

本书配套资源丰富, 有配套实验和题解, 需要者请登录人民邮电出版社教学服务与资源网 (<http://www.ptpedu.com.cn>) 免费下载。

本书可作为计算机及相关专业的本科教材, 同时也可作为自学参考用书和考试复习用书。

未经许可, 不得以任何形式和手段复制或抄袭本书任何内容。

21 世纪高等学校计算机规划教材

操作系统教程

-
- ◆ 编 著 孟 静 唐志敏
责任编辑 滑 玉
执行编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 17
字数: 410 千字 2009 年 4 月第 1 版
印数: 1—3 000 册 2009 年 4 月河北第 1 次印刷

ISBN 978-7-115-19524-1/TP

定价: 29.80 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

有需要者请登录人民邮电出版社教学服务与资源网 (<http://www.ptpedu.com.cn>) 免费下载。

衷心感谢上海交通大学尤晋元教授为本书作序。衷心感谢张尤腊老师为本书所提供的帮助。

感谢孟宪涛先生、李玉霞女士、孟庆伍先生为本书部分章节所做的资料收集和整理工作。

由于作者水平有限，书中疏漏之处在所难免，敬请读者不吝赐教。

作 者

2008 年秋于中关村

2.3.2	进程模型概述	47
2.3.3	用户如何让几个进程并发运行：多窗口，后台符，fork	47
2.3.4	进程三态转换分析	47
2.3.5	进程模型实现机制	50
2.3.6	专题：栈、程序文件结构与进程空间（进程映像）结构	52
2.3.7	专题：可抢先、不可抢先、完全可抢先	54
2.3.8	专题：进程调度算法	55
2.4	Linux 进程模型	56
2.4.1	Linux 进程管理用户界面	56
2.4.2	Linux 进程模型功能特点和实现机制	61
2.4.3	Linux 初始过程和进程层次	62
2.4.4	Linux 进程表和任务结构	63
2.4.5	Linux 进程状态	64
2.4.6	Linux 中断处理机制	66
2.4.7	Linux 进程调度算法	67
2.4.8	例析：Shell 和 fork 的内部工作过程	69
2.5	多任务模型之二：线程模型简介	71
2.6	Linux 线程管理简介	76
*2.7	作业管理	76
	习题 2	81
第 3 章	内存管理	82
3.1	内存管理概述	82
3.1.1	内存概念、作用、性能指标和计算机存储层次	82
3.1.2	内存硬件接口使用特性：微观角度（指令级）和宏观角度（程序级）	84
3.1.3	用户（程序）对内存的使用要求	88
3.1.4	内存管理的功能和任务	89
3.2	连续模式	92
3.2.1	无管理模式、覆盖技术和动态装入技术	92
3.2.2	单一分区模式和交换技术	94
3.2.3	固定分区模式和多道技术	96
3.2.4	可变分区模式和动态存储分配技术	98
3.3	不连续模式之一：页模式	101

3.3.1	实存页模式的基本工作过程与结构	102
3.3.2	虚存页模式的基本工作过程和结构	103
3.3.3	页式实现专题讨论(1): 虚存概念和作用	105
3.3.4	页式实现专题讨论(2): 进程页表的实现——快表、页表页和页目录	105
3.3.5	页式实现专题讨论(3): 大而稀疏内存使用	110
3.3.6	页式实现专题讨论(4): 页分配策略——请求调页、预先调页和写时复制	112
3.3.7	页式实现专题讨论(6): 页淘汰策略、工作集理论和颠簸	113
3.3.8	页模式评价、实际系统采用情况和本节小结	114
3.4	不连续模式之二/三: 段模式和段页式	115
3.4.1	段模式	115
3.4.2	段页式	120
3.5	Linux 内存管理	122
3.6	本章总结	124
3.6.1	*内存管理概念总结模型: 四空间模型	124
3.6.2	各模式的比较	125
	习题 3	127
第 4 章	外存管理和文件系统	129
4.1	外存管理和文件系统概述	130
4.1.1	外存硬件接口特性	130
4.1.2	用户对外存的使用要求	135
4.1.3	从文件定义看文件系统的界面高度和工作任务	136
4.2	文件系统用户界面	140
4.2.1	文件级界面: 文件属性和文件操作	140
4.2.2	目录级界面: 目录(树)和链接	142
4.2.3	文件子系统级用户界面	145
4.3	文件的实现	150
4.3.1	连续分配背景下的讨论	151
4.3.2	不连续分配背景下的讨论	156
4.3.3	各种分配策略的总结比较和综合采用	161
4.4	目录的实现	162
4.4.1	目录树结构的实现: 目录文件方法	162
4.4.2	硬链接的实现: 设备目录与文件目录的分离	164

4.4.3	符号链接的实现	167
4.5	文件子系统的实现	168
4.6	文件系统性能改善机制	171
4.6.1	物理地址与存取单位的优化	171
4.6.2	文件打开与关闭技术	172
4.6.3	文件共享	173
4.6.4	内存缓冲区与缓冲池	175
4.6.5	磁臂调度技术	176
4.7	Linux 文件系统	176
4.7.1	Linux 文件系统用户界面	177
4.7.2	文件系统类型无关层——虚拟文件系统 VFS	183
4.7.3	文件系统类型相关层——重点简介 Ext2 文件系统	183
4.7.4	mmap 系统调用	184
4.8	本章总结和有关的文件系统模型	189
	习题 4	190
第 5 章	设备管理	192
5.1	设备管理概述	193
5.1.1	计算机外部设备的定义与分类	193
5.1.2	设备硬件接口特性	195
5.1.3	用户对设备的使用要求	201
5.1.4	操作系统设备管理功能的任务	203
5.2	Linux/UNIX 设备管理实例分析	204
5.2.1	Linux/UNIX 设备管理用户界面	204
5.2.2	Linux/UNIX 设备管理内部实现过程和结构	206
5.3	设备管理界面和原理通述	211
5.3.1	操作系统设备管理用户界面通述	211
5.3.2	操作系统设备管理功能的内部实现结构	212
5.3.3	操作系统设备管理功能的内部工作过程	214
*5.3.4	速度匹配专题讨论(1): 设备完成技术、同步和异步 I/O	215
*5.3.5	速度匹配专题讨论(2): 缓冲技术	217
*5.3.6	设备分配与共享技术专题讨论: 独占、共享和虚拟设备	219
*5.3.7	速度匹配专题讨论(3): 联机、脱机和假脱机技术	220

*5.3.8 非程式 I/O 技术专题讨论: DMA、通道等	221
*5.3.9 设备驱动程序与核心间的接口规范例析: UNIX DDI/DKI 标准	224
5.3.10 驱动程序例析: 显示器、键盘、打印机驱动程序、电源	224
5.3.11 即插即用、驱动程序和设备安装与卸载、装入、启动、初始化、打开之间的 区别与关系	226
习题 5	226
第 6 章 进程通信	228
6.1 进程通信概述	228
6.2 进程互斥和同步机制	229
6.2.1 基本的硬件机制	229
6.2.2 软件忙等互斥方案	231
6.2.3 软件非忙等互斥方案: 信号量及其变种	233
*6.2.4 由程序设计语言支持的进程互斥机制: 管程	234
6.2.5 其他方案及其等价性	234
6.3 进程通信机制	235
6.4 死锁和饥饿	236
*6.5 Linux 进程通信	236
6.5.1 信号量、消息队列、共享内存	236
6.5.2 管道和有名管道	238
6.5.3 信号	239
习题 6	239
第 7 章 操作系统高级教程内容简介	241
7.1 分布式、网络、并行和嵌入式操作系统	241
7.1.1 计算机系统两极分化的发展趋势	241
7.1.2 分布式系统概述	242
7.1.3 并行操作系统	243
7.1.4 网络操作系统	244
7.1.5 分布式操作系统	245
7.1.6 机群与网格操作系统	248
7.1.7 嵌入式操作系统	249
7.2 操作系统性能评价	250

7.2.1 性能	250
7.2.2 可扩充性、可移植性、兼容性	251
7.2.3 安全性	252
7.2.4 可靠性和 RAS 技术	252
*7.2.5 低功耗	253
7.3 操作系统结构设计	255
7.3.1 单体结构模型	256
7.3.2 层次结构模型	256
7.3.3 客户/服务器模型（微核结构）	257
习题 7	257
参考文献	258

第 1 章

操作系统概论

读者肯定是带着许多问题开始读这本书的，这些问题可以概括为如下 6 个。

1. What—操作系统是什么？做什么？
2. Why—为什么需要操作系统？
3. How—操作系统如何工作？如何使用？
4. What—本课学习哪些内容？
5. Why—学了本课后有何用处？
6. How—如何学？

本书第 1 章就来解答这 6 个问题。对其中的第 3 个问题只做简要的解答，其系统、深入的答案构成全书的主要内容。其余 5 个问题的答案也需要读者在全书的学习过程中逐步加深了解和理解。实际上，在学习任何一门课程之前，都有类似的这 6 个问题。对任何一门课程的学习也都是对这 6 个问题的解答，课程主要内容也都是由第 3 个问题的答案构成的。

1.1 操作系统是什么与为什么

也许读者说不出来操作系统是什么，但只要用过计算机，肯定就使用过操作系统。现在把“操作系统是什么”这个问题放一放，先来看一看你用过的那（几）个操作系统。

1.1.1 引言：你所用过的操作系统

读者或多或少用过或听说过一些程序或软件。那么，下面这些软件中哪些是操作系统？极品飞车、Windows、Turbo C++、Word、Visual FoxPro、UNIX、用户自己编写的一个 C 语言源程序、Turbo-ASM、VI、Linux。

虽然有些读者不能概括地说出操作系统的定义，但能知道上述软件中只有 Windows、Linux 和 UNIX 是操作系统，其余的软件都是用户程序和其他系统软件（见表 1-1）。实际存在的操作系统有许多种（如果包括过去的就有几百种），目前最常用的就是上面这 3 种（UNIX 常见变种有 Solaris、AIX、HP UX 等，详见 1.5 节）。其他常用的还有 Mac OS（苹果计算机上的操作系统）、NetWare（Novell 公司的网络操作系统）、OS/2（IBM 为其高档个人计算机 PS/2 设计的操作系统）和 OS/400（IBM 小型机上的操作系统）等。

表 1-1 以下软件中哪些是操作系统

软件名	说明
极品飞车	游戏软件，属于应用软件
Windows	操作系统，是 IBM 个人计算机系列上最常用的窗口多任务操作系统
Turbo C++	编译程序，属于系统软件

软件名	说明
Word	编辑排版程序，是 Windows 系列操作系统下的应用软件
FoxPro	数据库管理程序
UNIX	操作系统，是多用户计算机上最常用的操作系统之一，其具体变种常见的有 SUN 公司的 Solaris、IBM 公司的 AIX、HP 公司的 HP UX 等
自己编写的一个 C 语言源程序	严格地说，这不算程序，只能算数据。这个源程序经编译、链接后产生的才是程序——可执行目标程序
Turbo-ASM	汇编程序，属于系统软件
VI	编辑程序，是 UNIX 操作系统下的应用软件
Linux	操作系统，其界面类似于 UNIX，是目前最常用的操作系统之一

根据使用经验（虽然可能是有限的），你能说一说自己用过的 Windows、Linux 和 UNIX 这些操作系统能做什么吗？

(1) 用计算机做任何事，都需先运行某个相应的程序。在 Windows 系列操作系统下，通常是通过双击一个程序的图标或程序名来运行该程序，这个图标或程序名通常出现在桌面、桌面的“开始”菜单、资源管理器等处。如果开机后不出现桌面，那就什么程序都不能运行。桌面是 Windows 操作系统显示的，程序图标的显示、双击鼠标动作的接收和翻译、启动执行相应的程序都属于 Windows 操作系统的功能。此外，在 Windows、Linux、UNIX 等操作系统下还可以通过命令方式来启动程序，这时也是由操作系统负责显示命令提示符、接收命令行、启动执行相应的程序。

(2) 不管将计算机用于哪个应用领域，都经常需要进行复制或删除文件、查看磁盘内容、建立文件夹等工作。在 Windows 操作系统下，可在资源管理器中通过菜单和鼠标操作来完成这些工作。资源管理器是 Windows 操作系统的一个部件，复制文件、查看磁盘内容等工作都属于 Windows 操作系统的功能。在 Linux 和 UNIX 操作系统下，这些常见工作还可以通过命令方式来完成，如 ls（查看目录内容）、cp（文件复制）、rm（删除文件）命令等，这些命令都是由操作系统实现的，即实现这些命令的程序代码都是操作系统代码的一部分。

(3) 在 IBM PC 上用汇编语言编写程序时，都需要 INT 语句。使用 INT 语句的目的大都是要做一些 I/O 工作，如文件读写或打印等。INT 语句实际上是一种特殊的调用语句，它调用的是相应操作系统（例如 Windows，如果用户是在 Windows 下编写汇编程序）的内部功能，通过这种方式，操作系统向用户程序提供帮助（详见 1.2.2 小节）。在 Linux 和 UNIX 操作系统中，也有类似的内部功能和相应的对外调用接口。

(4) 在 Windows 中，可以同时运行多个程序（即多任务）。例如，用户在等待一个文件下载（通过运行 IE 程序）的同时，可以编辑另一个文件（通过运行 Word 程序），这就提高了工作效率和机器利用率。多任务方式是 Windows 提供的。20 世纪 80 年代流行的 DOS 操作系统中就没有类似的功能，因此不能在 DOS 下同时运行两个程序，而 Linux 和 UNIX 还允许多个用户同时使用一台计算机，这就更进一步提高了机器利用率。

至此，读者已有了一个印象：操作系统的功能很丰富、很庞杂、很零散、很难概括，好像什么事都做。但操作系统又不可能什么事都能做，例如：操作系统不做天气预报，而是由专门的天气预报软件来做的；操作系统不做房屋设计，而是由专门的建筑 CAD 软件来做的；操作系统不是编译程序，用户使用什么语言编写的源程序就用对应该语言的编译程序……

总之，操作系统不直接解决具体的应用问题，也不负责编译源程序。那么，操作系统到底做什么，不做什么，它准确的功能范围是什么，它到底是一个怎样的软件，有没有统一的内在本质呢？

1.1.2 操作系统是什么与做什么

操作系统（operating system, OS）是计算机中最重要的系统软件，是这样一组系统程序的集成：这些系统程序在用户对计算机的使用过程中，即在用户程序运行中和用户操作过程中，负责完成所有与硬件因素相关的（硬件相关）和所有用户共需的（应用无关）基本工作，并解决这些基本工作中的效率和安全问题，为使用户（操作和上层程序）能方便、高效、安全地使用计算机系统，而从最底层统一提供通用的帮助和管理（见图 1-1）。

那么，在用户操作和用户程序中，哪些工作内容是硬件相关和应用无关的呢？主要是以下 4 个方面的工作，操作系统是完成以下 4 个方面工作的诸系统程序的集成（见表 1-2）。

（1）负责启动执行每个用户程序，并负责完成每个用户程序的结束处理工作，使每个用户程序可以很方便、灵活地启动执行和中止。

（2）在任何用户程序的运行过程中，负责完成所有的硬件相关和应用无关工作。每当用户程序运行过程中涉及这些工作，就通过一种特殊调用（称为系统调用）或中断方式，调用或进入操作系统来完成，从而为用户程序方便使用计算机系统，提供统一的帮助和管理。

（3）为用户对计算机进行基本操作提供现成的实用程序和相应的管理，以便这些操作能方便、有效地完成。这里所说的基本操作是指任何应用或开发背景下都通用和普遍需要、经常发生的操作，如复制文件、删除文件、显示磁盘目录内容或文件内容、格式化磁盘等。

（4）改善上述 3 方面工作中的效率和安全问题，使计算机系统的各个部分和整个计算机系统得到高效、安全的使用。

以上 4 个方面的工作看似互无关联，但它们具有共性：硬件相关和应用无关（见表 1-2）。反过来说，计算机使用过程中的所有硬件相关和应用无关工作就体现在这几个方面。

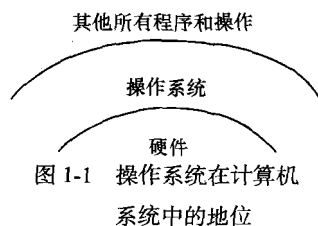


图 1-1 操作系统在计算机系统中的地位

表 1-2

操作系统的所有具体工作和它们的共性

	1	2	3	4
操作系统负责的所有工作有 4 个方面	负责启动执行每个程序并负责完成每个程序的结束处理工作	每当用户程序中使用 I/O 设备、所存储的信息、内存等时，其中的硬件相关和应用无关工作都通过调用操作系统来完成	为用户的常用基本操作提供现成程序。这里的常用基本操作是指任何应用都需要和通用的操作	解决以上工作中的安全和效率问题
这些工作的具体内容和完成方式举例	例如，通常启动一个程序只需在屏幕上双击相应图标即可，此外操作系统还提供命令方式、批处理、程序中启动等多种启动方式	例如，在个人计算机上的用户程序需打印字符时，只需以一条 INT 指令调用操作系统即可	例如，Windows 中提供了资源管理器等现成程序，供用户复制文件、查看目录等。用户需要复制文件时只需运行这些现成程序，不用自己编程	例如，操作系统提供多任务方式，可以提高效率

续表

		1	2	3	4
这些工作都与硬件相关	实现该工作的过程代码和硬件因素密切相关(即需要设置与测试、使用物理地址、设备物理接口寄存器等)	启动和结束过程涉及对 I/O 设备、内外存的使用与处理,需要物理地址与物理接口寄存器	例如,打印一个字符的过程中需要设置和查看打印机接口寄存器的每一位,需要了解和使用打印机接口的 I/O 物理地址	这些操作大量涉及对外存、I/O 设备的使用	例如,多任务方式的实现过程中需要管理内存和 I/O 设备
	硬件相关必然复杂、繁琐、代码量大	一个程序启动和结束处理工作所需的代码要与外存接口寄存器、外存物理地址、内存物理地址打交道,其代码量可能比程序主体代码量还大	用户程序对 I/O 设备的使用所涉及的代码经常占程序主体代码的大部分,这些代码中多处涉及接口寄存器、物理地址等复杂、繁琐的硬件细节	用户程序对 I/O 设备的使用所涉及的代码经常占程序主体代码的大部分。这些代码中多处涉及接口寄存器、物理地址等复杂、繁琐的硬件细节	例如,提供多用户方式的操作系统比不提供多用户方式的操作系统要大得多,复杂得多
	硬件相关的工作,其实现代码不通用,当硬件变化时,需要重新编写或编译相应代码	例如,如果程序从软盘移到磁带上,其启动代码就需要重新编写	例如,计算机系统换了一台不同型号的打印机,则程序中涉及打印输出的代码都要重新编写		
这些工作都与应用无关	任何应用(使用)都需要该工作	用计算机做任何事都需运行相应程序,而每个程序都需要启动和结束处理工作	任何程序的运行过程中都需要使用内存、I/O 设备、外存信息。只要使用 I/O 设备,就要与这些设备的接口寄存器打交道,只要使用内存,就要与物理地址打交道	不管是财务应用或是人口统计应用,任何应用中都需要进行这些工作	每个用户都希望提高使用效率并希望自己的数据有安全保障,每个机器主人则希望提高机器利用率
	在不同应用中,该工作的过程都是相同的	所有程序的启动过程都是雷同的(就同一种启动方式而言)。所有程序的结束处理工作也都雷同(就同一种结束原因而言)	例如,打印字符 A 与打印字符 E 的过程是一样的,只不过打印的数据(A 与 E)不同	复制一个财务总账文件和复制一个人口清单文件,其复制过程都是相同的,只不过复制的数据不同	
	与具体应用无直接关系(即与用户所关心的应用目标无直接关系)	例如,一个财务软件,其启动和结束处理工作同财务没有直接关系	财务软件中,打印一个数据的过程与财务逻辑上无直接联系	例如,复制的过程与复制的数据无关	
如果没有操作系统来完成这些工作,用户操作和用户程序会怎么样	所有程序都必须是由自启程序,即自包含引导装入代码,所有程序的启动操作都相当于一次关开机操作,麻烦、费时、不灵活	所有程序中都包含硬件相关代码,程序员必须了解相应硬件细节知识,编写大量复杂代码。当硬件变化时,必须重新编写程序	用户要自己编写程序实现这些常用操作	用户要自己解决所有安全和效率问题或不能解决	

续表

	1	2	3	4
由操作系统完成这些工作后,用户操作和用户程序会怎么样	用户程序中不必包含启动自己执行的引导装入代码,不必包含与本程序具体功能无关的异常结束处理代码。所有结束出口处只需一条“返回系统”指令。用户不必考虑程序启动与结束,不必了解相关硬件细节知识,不必频繁重复编写相应代码	用户程序中不包含硬件相关及应用无关代码,即不包含与接口寄存器、物理地址打交道的代码。用户不必了解相关硬件细节知识和编写相应的复杂繁琐代码	用户不必自己编写程序实现这些常用操作	用户不必考虑硬件相关、应用无关的安全和效率问题

何谓硬件相关和应用无关?所谓一个(或一段)程序是硬件相关的,是指该程序代码中包含内外存及设备的物理地址,包含对设备接口寄存器和设备接口缓冲区的直接读写等操作。硬件相关的代码必然随硬件的变化而变化。这样的硬件变化包括内外存物理存储空间大小的变化,程序和数据在内外存物理存储空间中存放位置的变化,设备数量和类型的变化等。(但不包括 CPU 指令集的变化,否则所有的程序就都是硬件相关的。)所谓操作系统承担硬件相关工作,使其上层的用户程序都是硬件无关的,是指当被用户程序使用的硬件发生变化(除 CPU 指令集外)时,用户程序不必改变,人的操作更不必改变,即使是 CPU 指令集发生改变时,所需做的改变工作也是最少的。

一个工作是应用无关的,是指不管用计算机来做什么,不管在计算机上运行什么程序,只要使用相应硬件或相应信息时就要涉及的工作。它是用户共需的,且工作过程都相同,有共性可循,却又与应用问题没有直接关系。

操作系统为用户的操作和程序完成所有硬件相关和应用无关的工作,目的和益处何在?硬件相关,必然意味着复杂繁琐、代码量很大、代码不通用和变化大,需要用户投入大量的精力来设计实现和维护修改,并了解相应的大量硬件细节知识,故有必要统一管理,使用户摆脱负担。应用无关,就意味着更有必要统一管理(因为普遍和频繁涉及,与具体应用无直接关系)和能够统一管理(因为工作过程相同)。越是计算机使用中底层的、基本的工作,越具有硬件相关和应用无关的特点,对用户和系统的方便、效率、安全影响越大,越需要、值得并可能由操作系统来完成,解决其中的效率和安全问题,从而使用户程序成为硬件无关的,即独立于硬件,不包括硬件相关的代码,硬件改变时程序不必改变。用户不必考虑这些底层基本使用过程和了解相关的硬件细节知识,避免频繁重复编写(或编译)与应用问题本身关系不大的大量复杂繁琐的代码,专心于应用本身,以更好地达到最终的具体应用要求和目标。总之,操作系统为保证用户的操作和用户程序最终使用的方便、效率、安全,承担了所有硬件相关和应用无关的工作,从最底层提供统一帮助和管理。

操作系统为用户完成所有硬件相关和应用无关的工作并不意味着操作系统本身的所有功能都是硬件相关的。操作系统分为实用程序层、命令解释层、核心层,其中只有核心层才是硬件相关的,甚至在核心(层)内部也做了进一步的隔离,只有核心层中最底层的一些模块(如 Windows 中的硬件抽象层、设备驱动程序等)才是硬件相关的。操作系统有狭义(核