

高等学校教材

操作系统原理简明教程

孟摇静摇编著

高等教育出版社

内 容 提 要

本书深入浅出、简明易懂地介绍了操作系统的原理和使用。全书共七章,第一章为操作系统概论,第二章~第六章依次讲述处理机管理、内存管理、外存管理和文件系统、设备管理和进程通信的原理,第七章介绍分布式、网络、并行和嵌入式操作系统,以及操作系统性能评价和结构设计技术。

作者所著的“面向 21 世纪课程教材”——《操作系统教程——原理和实例分析》出版后,受到全国许多高校老师的认可与好评。为了适应高校教学时数少以及非计算机专业对操作系统课程的教学要求,本书对原书中非重点、非主流实用的内容进行了删减。

本书既可作为高等学校计算机专业本、专科教材,也可作为非计算机专业的操作系统课程教材,同时也适合自学和考试复习使用。

目 录

第一章 操作系统概论	用户对处理机的使用要求和操作系统处理机管理功能的工作任务
1.1 操作系统是什么与为什么	1.1.1 进程模型
1.2 引言:你所用过的操作系统	1.1.2 进程三态转换分析
1.3 操作系统是什么与做什么	1.1.3 进程模型实现机制
1.4 操作系统的规模、数量与重要性	1.2 专题:可抢先、不可抢先、完全可抢先
1.5 操作系统如何工作	1.3 专题:进程调度算法
1.6 操作系统的第一工作:负责所有程序的启动和结束	1.4 进程模型实例分析(员)
1.7 操作系统的第二工作:用户程序中对操作系统的调用——系统调用和中断	1.5 关于建立进程和终止进程的用户界面
1.8 操作系统的第三工作:为常用基本操作提供现成实用程序	1.6 进程层次和初启过程
1.9 操作系统的第四工作:解决效率和安全问题——并发技术等	1.7 进程模型的基本结构和工作过程
1.10 从各种角度看操作系统	1.8 例析:和的内部工作过程
1.11 操作系统的结构	1.9 进程模型实例分析(员)
1.12 操作系统的接口	1.10 进程模型功能特点、用户界面和实现机制总瞰
1.13 操作系统的工作过程	1.11 初始过程和进程层次
1.14 操作系统的特点	1.12 进程表和任务结构
1.15 操作系统的类型	1.13 进程状态
1.16 操作系统的各种别名、比方和观点	1.14 中断处理机制
1.17 操作系统发展简史	1.15 进程调度算法
1.18 目前常用操作系统简介:等	1.16 线程模型简介
1.19 习题一	1.17 习题二
第二章 处理机管理	第三章 内存管理
2.1 处理机管理概述	3.1 内存管理概述
2.2 处理机硬件使用特性	

摇摇猿猿猿内存概念、作用、性能指标和计算机存储层次	摇摇猿猿猿不连续模式之二 页段模式和段页式
摇摇猿猿猿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)	摇摇猿猿猿段模式
摇摇猿猿猿用户(程序)对内存的使用要求	摇摇猿猿猿段页式
摇摇猿猿猿内存管理的功能和任务	摇摇猿猿猿内存管理实例分析
摇摇猿猿猿连续模式	摇摇猿猿猿幸幸幸内存管理
摇摇猿猿猿无管理模式、覆盖技术和动态装入技术	摇摇猿猿猿猿猿内存管理
摇摇猿猿猿单一分区模式和交换技术	摇摇猿猿猿本章总结
摇摇猿猿猿固定分区模式和多道技术	摇摇猿猿猿内存管理概念总结模型:四空间模型
摇摇猿猿猿可变分区模式和动态存储分配技术	摇摇猿猿猿各模式比较
摇摇猿猿猿不连续模式之一 页模式	摇摇猿猿猿摇习习题三
摇摇猿猿猿实存页模式的基本工作过程与结构	第四章摇外存管理和文件系统
摇摇猿猿猿虚存页模式的基本工作过程与结构	摇摇猿猿猿外存管理和文件系统概述
摇摇猿猿猿页式实现专题讨论(员) 虚存概念和作用	摇摇猿猿猿外存硬件接口特性
摇摇猿猿猿页式实现专题讨论(圆) 进程页表的实现——快表、页表页和页目录	摇摇猿猿猿用户对外存的使用要求
摇摇猿猿猿页式实现专题讨论(猿) 大而稀疏内存使用	摇摇猿猿猿从文件定义看文件系统的界面高度和工作任务
摇摇猿猿猿页式实现专题讨论(源) 页分配策略——请求调页、预先调页和写时复制	摇摇猿猿猿文件系统用户界面
摇摇猿猿猿页式实现专题讨论(缘) 页长和页簇化	摇摇猿猿猿文件级界面:文件属性和文件操作
摇摇猿猿猿页式实现专题讨论(远) 页淘汰策略、工作集理论和颠簸	摇摇猿猿猿目录级界面:目录(树)和链接
摇摇猿猿猿页式实现专题讨论(苑) 盘交换区管理	摇摇猿猿猿文件子系统级用户界面
摇摇猿猿猿页模式评价、实际系统采用情况和本节小结	摇摇猿猿猿文件的实现
	摇摇猿猿猿连续分配背景下的讨论
	摇摇猿猿猿不连续分配背景下的讨论
	摇摇猿猿猿各种分配策略的总结比较和综合采用
	摇摇猿猿猿目录的实现
	摇摇猿猿猿目录树结构的实现:目录文件方法
	摇摇猿猿猿硬链接的实现:设备目录与文件目录的分离
	摇摇猿猿猿符号链接的实现
	摇摇猿猿猿文件子系统的实现
	摇摇猿猿猿文件系统性能改善机制

摇摇猿猿猿物理地址与存取单位的优化	摇摇猿猿猿非编程猿的技术专题讨论 :猿猿
摇摇猿猿猿文件打开与关闭技术	摇摇猿猿猿通道等
摇摇猿猿猿文件共享	摇摇猿猿猿设备驱动程序
摇摇猿猿猿内存缓冲区和缓冲池	摇摇猿猿猿习题五
摇摇猿猿猿磁臂调度技术	摇摇猿猿猿第六章猿进程通信
摇摇猿猿猿文件系统实例分析	摇摇猿猿猿猿进程通信概述
摇摇猿猿猿文件系统	摇摇猿猿猿猿进程互斥和同步机制
摇摇猿猿猿猿文件系统	摇摇猿猿猿猿基本的硬件机制
摇摇猿猿猿猿文件系统	摇摇猿猿猿猿软件的忙等互斥方案
摇摇猿猿猿本章总结和有关文件系统模型	摇摇猿猿猿猿软件非忙等互斥方案 :信号量及其变种
摇摇猿猿猿习题四	摇摇猿猿猿猿由程序设计语言支持的程序互斥机制 :管程
摇摇猿猿猿第五章猿设备管理	摇摇猿猿猿猿其他方案及其等价性
摇摇猿猿猿猿设备管理概述	摇摇猿猿猿猿进程通信机制
摇摇猿猿猿猿计算机外部设备的定义与分类	摇摇猿猿猿猿死锁和饥饿
摇摇猿猿猿猿设备硬件接口特性	摇摇猿猿猿猿进程通信实例分析
摇摇猿猿猿猿用户对设备的使用要求	摇摇猿猿猿猿猿文件系统进程通信
摇摇猿猿猿猿操作系统设备管理功能的任务	摇摇猿猿猿猿猿文件系统进程通信
摇摇猿猿猿猿设备管理实例分析	摇摇猿猿猿猿猿文件系统进程通信
摇摇猿猿猿猿设备管理界面和原理通述	摇摇猿猿猿猿猿习题六
摇摇猿猿猿猿操作系统设备管理用户界面通述	摇摇猿猿猿猿猿第七章猿进一步的学习内容
摇摇猿猿猿猿操作系统设备管理内部结构与过程通述	摇摇猿猿猿猿猿操作系统性能评价
摇摇猿猿猿猿速度匹配专题讨论(猿) :设备完成技术、同步和异步	摇摇猿猿猿猿猿操作系统结构设计
摇摇猿猿猿猿速度匹配专题讨论(圆) :缓冲技术	摇摇猿猿猿猿猿现代操作系统的两极分化
摇摇猿猿猿猿设备分配与共享技术专题讨论 :独占、共享和虚拟设备	摇摇猿猿猿猿猿分布式系统概述
摇摇猿猿猿猿速度匹配专题讨论(猿) :联机、脱机和假脱机技术	摇摇猿猿猿猿猿并行操作系统
	摇摇猿猿猿猿猿网络操作系统
	摇摇猿猿猿猿猿分布式操作系统
	摇摇猿猿猿猿猿机群与网格操作系统
	摇摇猿猿猿猿猿嵌入式操作系统
	摇摇猿猿猿猿猿习题七
	摇摇猿猿猿猿猿参考文献

第一章 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几个:

猿媛幸澡戴—操作系统是什么?做什么?摇摇摇摇源媛幸澡戴—本书学习哪些内容?

圆媛幸澡戴—为什么需要操作系统?

缘媛幸澡戴—学了本书后有何用处?

猿媛幸澡戴—操作系统如何工作?如何使用?

远媛幸澡戴—如何学?

本书第一章就来解答这几个问题。对其中的第三个问题只做大致的解答,其系统、深入的答案构成全书的主要内容。其余几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,在学任何一门课程之前,都有类似的这几个问题。对任何一门课程的学习也都是对这几个问题的解答,课程主要内容也都是第三个问题。

猿媛幸澡戴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,就肯定用过操作系统。现在先把“操作系统是什么”这个问题放一放,先来看一看用过的那(几)个操作系统。

猿媛幸澡戴引言: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、宰媛幸澡戴栽媛幸澡戴说且、宰媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴用户自己编写的一个“悦语言源程序、栽媛幸澡戴源媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴”

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宰媛幸澡戴栽媛幸澡戴说且和栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴是操作系统,其余的软件都是用户程序和其他系统软件(表媛幸澡戴)。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),目前最常用的是上面这猿媛幸澡戴栽媛幸澡戴说且(栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴常见变种有栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴等,详见猿媛幸澡戴栽媛幸澡戴说且)。其他常用的还有栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴(苹果计算机上的操作系统)、栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴公司的网络操作系统)、栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴为其高档个人计算机栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴设计的操作系统)和栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴小型机上的操作系统)等。

根据使用经验(虽然可能是有限的),你能说一说自己用过的宰媛幸澡戴栽媛幸澡戴说且和栽媛幸澡戴灾媛幸澡戴造媛幸澡戴操媛幸澡戴裁媛幸澡戴这些操作系统能做什么吗?

(员)用计算机做任何事,都需先运行某个相应的程序。在宰媛幸澡戴栽媛幸澡戴说且系列操作系统下,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 宰圣世操 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 宰圣世操 操作系统的功能。此外, 宰圣世操 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、接受命令行、启动执行相应的程序。

表 员 摇摇 以下软件中哪些是操作系统

软件件播名	说播播明
极品飞车	游戏软件, 属于应用软件
宰圣世操	操作系统, 是 陨运 个人计算机系列上最常用的窗口多任务操作系统
裁颢颢说垣	编译程序, 属于系统软件
宰燠燠	编辑排版软件, 是 宰圣世操 系列操作系统下的编辑软件
云颢颢颢	数据库管理软件
裁颢颢	操作系统, 是多用户计算机上最常用的操作系统, 其具体变种常见的有 裁颢颢公司的 裁颢颢 陨运公司的 颢颢 匀孕公司的 匀孕 裁颢颢等
自己编写的一个 悦语言源程序	严格地说, 这不算程序, 只能算数据。这个源程序经编译、连接后产生的才是程序——可执行目标程序
裁颢颢颢颢颢颢	汇编程序, 属于系统软件
灾颢	编辑软件, 是 裁颢颢 下的编辑软件
颢颢颢颢	操作系统, 其界面类似于 裁颢颢, 是目前最常用的操作系统之一

(圆) 不管将计算机用于何种应用领域, 都经常需要进行文件复制或删除、磁盘内容查看、建立文件夹等工作。在 宰圣世操 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 宰圣世操 操作系统的一个部件, 即文件复制、磁盘内容查看等工作都属于 宰圣世操 操作系统的功能。在 颢颢颢 和 裁颢颢 操作系统下, 这些常见工作还可以通过命令方式来完成, 例如 颢颢 查看目录内容)、颢颢 文件复制)、颢颢 (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。

(猿) 在 陨运 孕 说上用汇编语言编写程序时, 都需要 颢颢 语句。使用 颢颢 语句的目的大都是要做一些 颢颢 的工作, 比如文件读写或打印等。颢颢 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 宰圣世操 译 如果用户是在 宰圣世操 译 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 员颢颢 颢颢)。在 颢颢颢 和 裁颢颢 操作系统中, 也有类似的内部功能和相应的对外调用接口。

(源) 在 宰圣世操 译 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 陨 程序)的同时, 可以编辑另一个文件(通过运行 宰燠燠 程序), 这就提高了工作效率和机器利用率。多任务方式是 宰圣世操 译 提供的。圆颢颢 世纪 颢颢 年代流行的 颢颢 杂操作系统中就没有类似的功能, 因此不能在 颢颢 杂下同时运行两个程序, 而 颢颢颢 和 裁颢颢 还可以让

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。

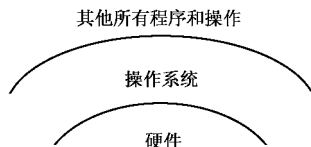
至此,读者已有了一个印象:操作系统的功能很丰富、很庞杂、很零散,很难概括,好像什么事都做。但操作系统又不可能什么事都做,例如,操作系统不做天气预报,这是由专门的天气预报软件来做的;操作系统不做房屋设计,这是由专门的建筑绘图软件来做的;操作系统不是编译程序,用户使用什么语言编写的源程序就用对应该语言的编译程序……

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?

猿猴操作系统是什么与做什么

操作系统(猿猴操作系统)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(图猿猴)。



图猿猴操作系统在计算机系统
中的地位

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?主要是以下四个方面的工作。操作系统是完成以下四个方面工作的诸系统程序的集成(表猿猴):

(猿) 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。

(圆) 在任何用户程序的运行过程中,负责完成所有硬件相关和应用无关的工作。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。

(猿) 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,例如复制文件、删除文件、显示磁盘目录内容或文件内容、格式化磁盘等。

(源) 改善上述三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。

以上四方面工作似乎互无关联,但它们都具有这样的共性:硬件相关和应用无关(见表猿猴)。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这个方面。

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,等等(但不包括 悦载指令集的变化,否则所有的程序就都是硬件相关了)。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 悦载指令集外)时,用户程序不必改变,人的操作更不必改变,即使是 悦载指令集发生改变时,所需做的改变工作也是最少的。

表 员猿 操作系统的所有具体工作和它们的共性

	员	圆	猿	源
操作系统负责的所有工作有四个方面	负责启动每个程序执行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到 障的设备、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命令提示符后输入程序名即可,此外操作系统还提供批处理、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 障指令调用操作系统即可	例如,障杂中为文件复制提供的现成程序有 悦再 阅再 再等。用户需复制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
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、设备接口寄存器等)	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,需要了解和使用的打印机接口的 障物理地址	这些操作大量涉及对外存、障设备的使用	例如,多任务方式的实现过程中需管理内存和障设备
	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 障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得比提供多用户方式得多,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,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		

续表

	员	圆	猿	源	
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、磁盘的设备、外存信息。只要使用磁盘设备,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,每个机器主人者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 粤与打印字符 耘的过程是一样的,只不过打印的数据(粤与 耘)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动程序,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,编大量复杂代码。当硬件变化时,必须重编程序	用户要为这些常用操作自行编制程序	用户要自行解决所有安全和效率问题或不能解决	
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	用户不必为这些常用操作亲自编程	用户不必考虑硬件相关、应用无关的安全和效率问题	

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到的工作,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。使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,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,但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(例如,在通用操作系统中的硬件抽象层、设备驱动程序等)才是硬件相关的。操作系统有狭义(核心)与广义(至实用程序层,甚至更高层——产品捆绑)之分,也缘于此。

员源缘 操作系统的规模、数量与重要性

从操作系统的上述四个工作可以看出,操作系统必然是个庞然大物。实际上也正是如此。例如,在通用操作系统中,操作系统都是从光盘安装(或网上下载),需占用几百兆字节的外存。

从员缘年开始出现操作系统到其后的源多年间,在各种机器上实际运行的操作系统有几百个,它们中最小的有上万行代码,大的达几十万行(猿万行,员猿万)甚至几百万行,甚至更多。它们在实际运行时占用了大量的机器时空资源:占用了圆缘~猿缘%的悦哉时间,操作系统本身占内存空间多达几十兆字节,占用外存空间多达几百兆字节,等等。这些数字都说明了操作系统的规模庞大。另外,不同机器上的操作系统一般来讲是不同的,同一种机器上也可以有不同的操作系统(例如,在个人计算机上可以运行宰圣世普和蕴生普等操作系统),同一种操作系统可以在不同机器上有不同版本或变种(例如,裁源戴可以在阴云杂源等很多不同厂商的机器上运行),同一机器上的同一操作系统也有功能升级导致的不同版本(基本上一至五年推出一次新版),这些因素又都导致了操作系统的数量众多。

这些众多庞然大物的产生,不可避免地为用户各自开发,发展到厂家提供,从免费配给,发展到商品化和独立注册,它们的产生(实现)与维护支持使厂家耗资巨大。据员缘年的估计,每个厂家的每个操作系统花费圆缘万~员缘缘万美元,而当时一个厂家同时支持十几个操作系统(多个产品线)、每个操作系统支持圆个~猿个版本的情况并不少见。据员缘年的估计,过去缘年中所有厂家、用户的操作系统总费用为几百亿美元。圆缘年推出的宰圣世普耗资几十亿美元。这种庞然大物的使用和维护,也给用户带来了繁重的学习

负担和管理负担,用户机构每年要付出大笔的维护和培训费用(尤其在版本升级时)。

是什么导致人们花费如此多的财力和时间在操作系统上呢?因为操作系统带给用户方便、效率和安全,从而给厂家和用户带来效益,虽然为它付出的代价巨大,但与它带来的方便、效率、安全和效益相比,这种代价是值得的。所以,操作系统虽是令人头痛的庞然大物,但每台计算机都离不开它,凡是与计算机打交道的人(不管是厂家还是用户)也都离不开它。今天,从最小的个人计算机到最大的巨型机,操作系统无处不在,缺一不可。用户很少使用一台没有操作系统的机器,或者说很少不在操作系统的帮助下使用计算机,以至到了对操作系统反而熟视无睹的地步。计算机产生初期没有操作系统时对计算机使用的艰难、低效已经成了遥远而陌生的事情。

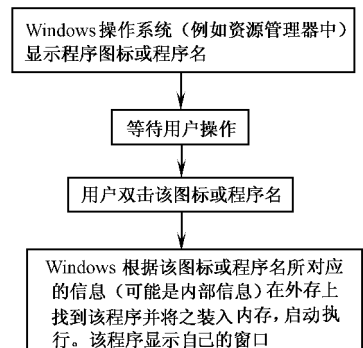
员程操作系统如何工作

员程操作系统的第一个工作:负责所有程序的启动和结束

摇摇用计算机做任何事,都需先运行某个相应的程序。那么程序如何开始执行呢?从硬件知识可知,任一(可执行目标)程序的启动执行有两个前提条件:一是该程序在内存中(该程序已装入内存),二是“**译**”中的程序计数器被置为该程序在内存的起始执行地址(这就意味着“**译**”执行的下一条指令是该程序的指令)。这样,程序很自然就开始运行了。那么,一个程序是由谁来装入和启动行的呢?关于“启动一个程序执行”的申请又是怎样提出的呢?解决这两个问题的答案形成了**远**种启动方式。

摇摇**译**程序的第一种启动方式——鼠标单击方式

宰宰**译**下,通常通过双击一个程序的图标或程序名来运行该程序。该图标或程序名通常出现在桌面、桌面上的“开始”菜单、资源管理器等处。程序图标或程序名的显示、双击动作的接收和翻译及找到并启动执行相应程序,都属于宰宰**译**操作系统的功能,通常由宰宰**译**中的资源管理器完成,其过程见图**员程**



在宰宰**译**中,一个正在运行的程序通常与一个**摇图**员程宰宰**译**操作系统中以鼠标方式启动程序的过程窗口相联系:启动了一个程序,就打开了一个相应窗口(虽然该窗口有时会被最小化或覆盖);关闭了这个窗口,就关闭了这个程序。

这种鼠标操作方式和窗口界面形象生动,操作简单而有规律,用户不必记忆命令,是当今所有操作系统中最常用的程序启动方式。**译**和**裁**操作系统中也提供了这种方式

(如 载原辛圣世册等)。

程序的第二种启动方式——命令方式

所有操作系统下,都提供了命令方式来启动一个程序。

在 宰圣世册下,可单击“开始”→“运行”→输入程序名(及其所在路径)与参数来启动一个程序(姑且将这种启动方式称为运行方式),也可以单击“开始”→“程序”→“配杂阔杂方式”→输入程序名(及其所在路径)与参数来启动一个程序(姑且称之为阔杂方式,阔杂是 愿世纪 愿年代 隔运 隔上最流行的操作系统)。运行方式和 阔杂方式都是命令方式。

除了 宰圣世册下的运行方式外,宰圣世册下的 配杂阔杂命令方式、蕴圣世册和 哉圣世册下的命令方式中都有命令提示符出现。命令提示符是操作系统在屏幕上向用户提供的一种提示标志,它的出现就表示“你可以在命令提示符后输入程序名来启动下一个程序了”。配杂阔杂方式下命令提示符的最常见形式是“悦跃”,蕴圣世册和 哉圣世册下命令提示符的最常见形式是“豫”或“裕等。通常把在命令提示符后输入的程序名及其参数称为命令行(到回车为止)或命令。任一命令行(即任一条命令)的本质都是申请某一程序执行。在命令方式下的用户工作过程见图 员藏彝。通常把操作系统中实现命令方式(以及批处理方式)的程序称为命令解释程序或命令解释器,其内部工作过程见图 员藏遭。

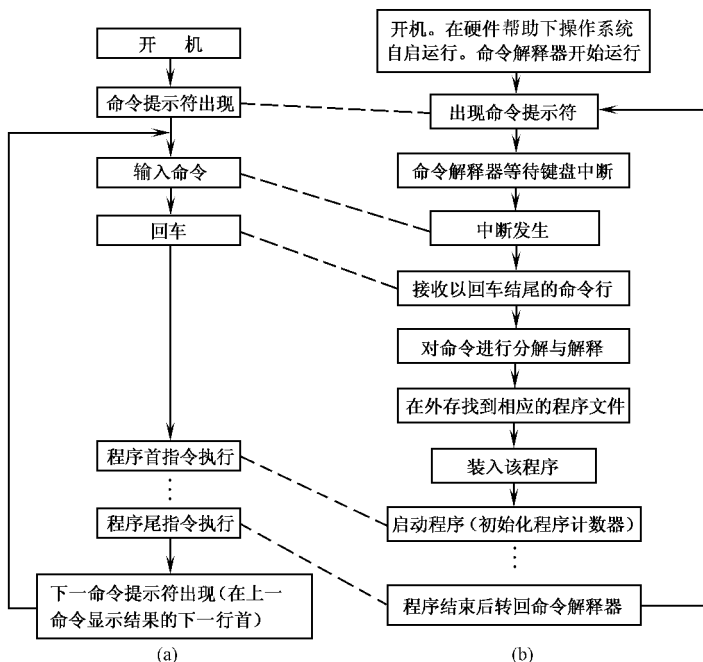


图 员藏彝 命令方式下的用户使用过程和内部实现过程

宰圣世册中 阅杂方式的命令解释器是 精自森世册,可以在 宰圣世册启动盘根目录下看到这个文件。故 粤戴中命令解释器通常称为 杂森造。同一操作系统可以有多个命令解释器,例如, 粤戴有 月匀、悦匀和 运匀等。

命令方式在 圆世纪 愿年代是启动程序运行最主要的方式,但如今它已经让位于鼠标单击方式而成为第二类常用的程序启动方式了。从作业管理角度说,鼠标单击方式与命令启动方式都属于交互式作业。

如果通过命令行启动的程序是操作系统提供的现成程序(即操作系统的第三个工作,参见 员程章节),则该程序、该命令(行)称为操作系统命令。

命令方式下,在命令提示符后输入一个命令而启动执行一个程序后,如何得知该程序是否运行完?是从下一个命令提示符的出现知道的。故命令提示符有两重含义:一是标志着上一个程序的结束(上一条命令的结束);二是标志着可以启动运行下一个程序(可以输入下一条命令)。

既然在开机通电状态的任一时刻都有程序在运行,或者说有指令在执行,那么在命令方式下,当命令提示符后为空(即光标紧挨在命令提示符后闪烁)而操作者长久未归时,计算机上是在运行程序吗?如果是的话,是什么程序?

从图 员猿可知,此时是负责命令的接受、解释和启动执行的命令解释器在运行。既然命令解释器负责接受、解释、执行命令行,而用户何时输入命令又是不确定的,因此为了在用户一旦输入命令后尽快地做出响应,命令解释器必须随时随刻准备接受命令,尤其是在命令提示符出现后(因为命令提示符出现的含义就是通知用户可以输入命令了)。所以,当命令提示符后为空时,表明命令解释器正在运行,即正在等待用户输入命令(例如通过一个循环来等待,这个循环不停地检测是否有键盘中断,即是否有用户开始输入命令)。

提问:① 为了以参数 孕执行程序 粤,而在命令提示符后输入一条命令“粤孕回车”,从开始输入命令到输入了命令尾的回车,直到程序 粤内的第一条指令执行前,计算机在做些什么?② 在命令方式下,从用户程序的最后一条指令执行完到下一命令提示符出现之间,计算机在做些什么?(这两个提问作为本章习题)

猿援程序的第三种启动方式——批方式

除鼠标方式、命令方式外,几乎所有操作系统都有对批文件的处理功能,在批文件中启动的程序称为以批方式启动,是程序启动的第三种方式。批方式是指将若干条命令(请注意命令即启动程序的请求)放在一个文件(即批文件)中,该批文件可以被启动执行(通过命令方式或鼠标方式),其执行过程就是由计算机自动连续执行该文件中的这组命令。批文件是传统的 阅杂中的说法(遭批文件),在 宰圣世册中,批文件称为脚本文件(泽批城还或 泽批城,在 蕴统普和 粤戴中,批文件称为 杂森造脚本文件(杂森造籍批城。在 蕴统普和 粤戴中,杂森造脚本文件的用途非常广泛,它实际上是一种“程序”(相应的“程序”设计语言称为 杂森造语言),用于把现有的程序(功能)组合起来实现更复杂的功能。一个典型的 杂森造脚本的内容如下(本 杂森造程序把每个 蕴统普命令的名字和功能汇集在一个文件中):

序等)还有很多类似的在一个程序中启动运行另一程序的情况和需要,甚至还有不必用户输入程序名,便直接根据需要(根据预先存储的程序名或计算出来的程序名)在一个程序中启动运行另一程序的。实际上,前述三种程序启动方式(鼠标、命令、批)最终都是通过这种“程序中启动”方式来实现的,参见图 员程圆

在一个程序中启动另一个程序,这种启动方式便于程序的灵活方便启动和动态自动启动,其外部使用过程和内部实现过程见图 员程

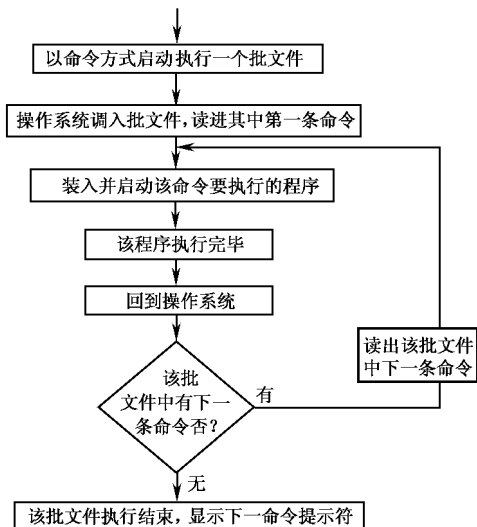


图 员程批方式下的工作过程

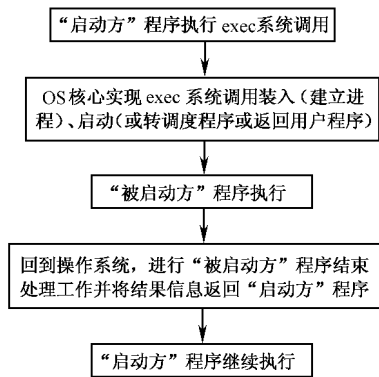


图 员程程序中启动方式的工作过程

提问:在一个程序中启动另一个程序的方式与子程序调用方式有何异同?(见本章习题)

缘程序的第五种启动方式——纯粹由硬件装入并启动程序执行

可以用硬件(即靠纯硬件功能)装入并启动程序执行吗?可以。最早期的计算机就是通过硬件装入并启动程序的(见图 员程)。用户先把装有可执行目标程序的纸带或卡片安装到纸带输入机或卡片输入机上,然后按一下机器面板上一个特定的“装入程序并启动执行”按钮(注意在此之前内存中什么也没有),硬件就从头开始把纸带上的内容顺序读到内存中,直至纸带上的一个特殊的程序结束标记为止,然后硬件开始从内存零地址处执行程序。这样的顺序连续装入过程是一个很机械的工作,硬件可以承受这样的复杂度,而且只要每个程序都从纸带头开始(即一个纸带一个程序),且都用硬件规定的结束标记,则每个程序的装入过程都是雷同的,用硬件装入也就没有通用性方面的问题。

但后来就很少用硬件装入及启动程序了,这是为什么呢?原因如下:

(员) 纯硬件装入的前提是程序必须在一个存储介质上从头开始顺序连续存放,这就限制一个存储介质上只能放一个程序,而且必须从头开始顺序连续存放。在早期只有纸带或