

● 高等学校21世纪计算机教材

现代

计算机操作系统

陈应明 李俊宇 梁盛伦 编著

冶金工业出版社

高等学校 21 世纪计算机教材

现代计算机操作系统

陈应明 李俊宇 梁盛伦 编著

北 京

冶金工业出版社

2004

内 容 简 介

操作系统是计算机系统中的一个重要的系统软件。本书从操作系统的基本概念和基本功能讲起,着重介绍了操作系统的基本工作原理、实现技术以及几种流行的操作系统。主要包括处理机管理、并发进程与死锁问题、存储管理、文件管理、设备管理、分布式操作系统概念和系统安全等内容。此外,为了加深大家对操作系统基本原理的理解,最后通过实例介绍了当今比较流行和实用的操作系统在实际工作中的应用。

本书针对性强,结构清晰,并且有一定深度。里面含有大量的示例图和示例代码,方便读者进行学习,同时力图反映操作系统技术的新成果和新发展。

本书既可作为大、中专院校计算机应用基础类教程,也可供各类计算机基础教学的培训班和自学人士使用。

图书在版编目(CIP)数据

现代计算机操作系统 / 陈应明等编著. —北京:冶金工业出版社, 2004.1

ISBN 7-5024-3451-8

I. 现... II. 陈... III. 操作系统 IV. TP316

中国版本图书馆 CIP 数据核字(2003)第 120107 号

出版人 曹胜利(北京沙滩嵩祝院北巷 39 号,邮编 100009)

责任编辑 戈兰

湛江蓝星南华印务公司印刷;冶金工业出版社发行;各地新华书店经销

2004 年 1 月第 1 版,2004 年 1 月第 1 次印刷

787mm×1092mm 1/16; 27 印张; 625 千字; 422 页; 1-5000 册

49.00 元

冶金工业出版社发行部 电话:(010) 64044283 传真:(010) 64027893

冶金书店 地址:北京东四西大街 46 号(100711) 电话:(010) 65289081

(本社图书如有印装质量问题,本社发行部负责退换)

前 言

一、关于操作系统

操作系统是配置在计算机硬件上的第一层软件，是对硬件系统的第一次扩充，它在计算机系统中占据了极其重要的位置。计算机操作系统的不断发展和完善，对计算机在大众化方面的应用中功不可没。其他所有软件，如：汇编程序、数据库管理系统、游戏软件等，都依赖操作系统的支持，都需要它的服务。

操作系统已经成为大型机、小型机乃至微型机、计算机网络都必须配置的软件。如果要让用户去使用一台没有配置操作系统的现代计算机，其中困难是难以想象的。操作系统控制和管理整个计算机系统中多种硬件、软件资源，并为用户提供一个方便灵活、安全可靠的工作环境。

学习操作系统要掌握其基本概念、基本功能、基本思想和基本算法，并学会利用这些知识去分析、解决一些理论问题和指导计算机操作，为以后学习和理解新型的操作系统，应用操作系统和在操作系统上开发自己的应用程序打下基础。

二、本书的内容结构

本书是作者根据《操作系统大纲》，再结合国内外优秀的操作系统系列教材，查阅了大量的资料写成的。全书围绕操作系统能做什么、怎么去做和为什么要这样做这三个问题展开论述。全书的结构如下：

第1章：计算机操作系统概述。主要介绍了操作系统的基本概念、计算机系统结构及操作系统结构，包括成分、功能、调用、程序、结构等。

第2章：进程与线程。主要介绍了进程和线程的概念、进程调度、控制以及其之间的通信、多线程的优点、用户线程和内核线程、多线程模式、Solaris 2 线程以及 Java 线程等。

第3章：处理机调度。主要介绍了调度的概念、标准及算法、进程控制、进程调度、调度算法以及死锁等内容。

第4章：存储管理。主要介绍了存储器和虚拟存储器的结构和工作原理。讲述了分区管理、分页管理、分段管理、段页式管理等方法 and 特点以及虚拟内存的相关知识。

第5章：文件系统。在介绍了文件基本概念的基础上，详细介绍了文件访问方式、目录结构、文件存储、文件的保护与保密及文件的使用等内容。

第6章：I/O 系统。主要介绍了 I/O 的硬件、数据传送控制方式、I/O 子系统核心、磁盘结构、磁盘的调度和管理以及其可靠性等内容。

第7章：分布式通信。主要讲述了分布式操作系统中的通信问题。包括背景、通信、通信协议、套接字编程、远程方法调用以及 CORBA 等。

第8章：分布式管理。着重讲述分布式系统中的设备以及各类资源的管理，包括事件排序、互斥现象、死锁处理、选举算法等。

第9章：系统安全问题。主要介绍在现今计算机系统和通信系统中计算机各类资源所

面临的各种威胁，包括入侵者、恶意软件等，以及如何去解决这些问题，其中涉及加密技术的应用。

第 10 章：UNIX 操作系统。以 UNIX 系统为实例，讲述了 shell 命令、进程管理、内存管理、文件系统、I/O 系统等。

第 11 章：Linux 操作系统。以 Linux 系统为实例，讲述了 Linux 的历史、进程管理、文件系统及跨进程通信等方面的知识。

第 12 章：Windows 2003 操作系统。以 Windows 2003 系统为实例，讲述内容包括系统结构框架、执行体、外部环境子系统、Windows 2003 文件系统及编程接口等。

此外，本书的最后还附有 Java 语言基础的附录，可以帮助对 Java 陌生的读者快速地了解本书中出现的大量 Java 语言的代码；附有习题参考答案，以方便读者对照课后的习题进行练习。

本书是按 80 学时编写的。

三、本书的特点

本书条理清晰，用简练的语言结合大量的图片、表格，详细地介绍了操作系统的基本原理和应用知识，并通过常用的操作系统实例总结了全书的知识要点。里面对一些重要的算法还给出了 Java 的代码。通过本书的学习，读者可以很快掌握操作系统的相关知识。

此外，本书每章后都提供了丰富的习题，读者可参照书后的参考答案进行练习，以巩固所学的知识。

四、本书的适用对象

本书既可作为大、中专院校计算机应用基础类教程，也可供各类计算机基础教学的培训班和自学人士使用。

读者如有好的意见或建议可发 E-mail 至：service@cnbook.net，也可登录 <http://www.cnbook.net>，在该网站的相关论坛进行探讨。

本书由陈应明、李俊宇、梁盛伦编写，参加本书编写的还有梁宇凌、李文，并对本书的编写提出了大量宝贵意见，在此编者对他们表示深深的谢意。

由于时间仓促，水平有限，书中不妥之处在所难免，敬请各位读者斧正。

编 者

2003 年 11 月

目 录

第1章 计算机操作系统概述	1	2.3.1 进程创建与撤消	41
1.1 操作系统简介	1	2.3.2 进程切换	43
1.1.1 什么是操作系统	1	2.3.3 上下文切换	43
1.1.2 批处理操作系统	3	2.4 交互进程	44
1.1.3 分时操作系统	5	2.5 进程间通信	46
1.1.4 实时操作系统	5	2.5.1 消息传递系统	47
1.1.5 分布式操作系统	6	2.5.2 消息传递通信的实现方法	47
1.2 计算机系统结构	7	2.5.3 同步	48
1.2.1 计算机硬件发展史	7	2.5.4 消息格式	49
1.2.2 CPU	8	2.5.5 缓冲	49
1.2.3 计算机运行原理	11	2.5.6 生产者——消费者示例	50
1.2.4 I/O 结构	12	2.6 线程的基本概念	50
1.2.5 存储结构	13	2.7 多线程的优点	51
1.2.6 硬件保护	18	2.8 用户线程和内核线程	52
1.3 操作系统结构	19	2.8.1 内核线程	52
1.3.1 系统成分	19	2.8.2 用户线程	52
1.3.2 系统功能	23	2.9 多线程模式	52
1.3.3 系统调用	25	2.9.1 单对多模式	52
1.3.4 系统程序	27	2.9.2 单对单模式	53
1.3.5 系统结构	28	2.9.3 多对多模式	53
1.3.6 虚拟机	32	2.10 Solaris 2 线程	53
1.3.7 Java	32	2.11 Java 线程	55
小结	34	2.11.1 线程的创建	55
综合练习一	35	2.11.2 线程管理	56
一、选择题	35	2.11.3 线程的状态	56
二、填空题	36	2.11.4 多线程解决示例	57
三、简答题	36	小结	59
四、应用题	36	综合练习二	59
第2章 进程与线程	37	一、选择题	59
2.1 进程的基础	37	二、填空题	60
2.1.1 进程的概念	37	三、简答题	60
2.1.2 进程状态	37	四、应用题	60
2.1.3 进程控制块	38	第3章 处理机调度	61
2.2 进程调度	39	3.1 调度的基本概念	61
2.2.1 进程调度队列	39	3.1.1 处理机调度的功能	61
2.2.2 调度程序	40	3.1.2 处理机调度的时机	62
2.3 进程控制	41	3.2 调度标准	62
		3.3 调度算法	63

3.3.1 先来先服务调度算法	64	综合练习三	110
3.3.2 最短作业优先法	65	一、选择题	110
3.3.3 优先级数法	65	二、填空题	111
3.3.4 轮转法	67	三、简答题	111
3.3.5 多级反馈队列调度算法	68	四、应用题	111
3.4 多处理机调用	69	第4章 存储管理	113
3.4.1 多处理机系统的基本概念	69	4.1 背景知识	113
3.4.2 多处理机操作系统的分类	70	4.1.1 地址绑定	113
3.4.3 多处理机系统调度策略	71	4.1.2 地址变换	114
3.5 实时调度	73	4.2 分区管理及相关技术	115
3.5.1 实现实时调度的基本条件	73	4.2.1 分区管理	116
3.5.2 实时调度算法的分类	74	4.2.2 配置算法	118
3.5.3 常用的几种实时调度算法	75	4.2.3 覆盖技术	119
3.6 Java 线程调度	77	4.2.4 交换技术	120
3.6.1 时间片限制	77	4.3 页式管理	121
3.6.2 线程优先级	78	4.3.1 简单分页	122
3.6.3 基于 Java 的轮转法调度程序	78	4.3.2 辅助寄存器与快表	123
3.7 并发进程	80	4.3.3 页表保护	125
3.8 临界资源和临界区	81	4.3.4 多级分页	126
3.8.1 临界资源	81	4.3.5 反向页表	126
3.8.2 临界区	82	4.3.6 共享页	127
3.9 双作业的解决方案	82	4.3.7 页式管理的优缺点	128
3.10 硬件同步	85	4.4 段式管理	128
3.11 信号量	87	4.4.1 简单分段	129
3.11.1 实现	87	4.4.2 段式地址转换	131
3.11.2 用法	89	4.4.3 共享与保护	132
3.11.3 死锁与饿死	90	4.4.4 碎片管理	133
3.12 经典同步问题	90	4.4.5 段式管理的特点	133
3.12.1 有界缓冲区问题	90	4.5 段页式管理	134
3.12.2 读者和写者问题	93	4.6 虚拟内存背景知识	136
3.12.3 哲学家就餐问题	96	4.6.1 分页式虚拟存储	137
3.13 管程 (monitor)	97	4.6.2 分段式虚拟存储	138
3.13.1 管程的定义	97	4.7 虚拟内存请求页式管理	138
3.13.2 条件变量	98	4.7.1 基本概念	139
3.13.3 用管程机制解决哲学家就餐问题	99	4.7.2 请求页式管理工作性能	141
3.14 Java 同步机制	100	4.8 虚拟内存的页替换算法	143
3.15 死锁的定义	102	4.8.1 基本机制	143
3.15.1 死锁的描述	103	4.8.2 先入先出策略 (FIFO)	145
3.15.2 解决死锁的方法	106	4.8.3 理想页置换算法	147
3.15.3 死锁预防	106	4.8.4 LRU 算法	148
3.15.4 死锁避免	107	4.8.5 LRU 近似算法	149
3.15.5 死锁的检测与恢复	108	4.8.6 计数算法	151
小结	109		

4.8.7 页缓冲算法.....	151	小结.....	189
4.9 虚拟内存的页面分配.....	152	综合练习五.....	190
4.9.1 最小页面数.....	152	一、选择题.....	190
4.9.2 分配算法.....	153	二、填空题.....	191
4.9.3 全局替换与局域替换.....	154	三、简答题.....	191
4.10 抖动 (Thrashing)	154	四、应用题.....	191
4.10.1 抖动的原因	154	第 6 章 I/O 系统.....	192
4.10.2 局部性原理与工作集概念.....	155	6.1 引言.....	192
4.10.3 缺页率.....	157	6.2 I/O 硬件.....	193
小结.....	157	6.2.1 设备控制器.....	194
综合练习四.....	159	6.2.2 直接存储器访问.....	195
一、选择题.....	159	6.2.3 通道方式和输入/输出处理器.....	196
二、填空题.....	160	6.3 数据传送控制方式.....	198
三、简答题.....	160	6.3.1 程序方式.....	199
四、应用题.....	160	6.3.2 中断方式.....	200
第 5 章 文件系统.....	162	6.3.3 DMA 方式.....	204
5.1 文件系统的概念.....	162	6.3.4 通道控制方式.....	207
5.1.1 文件属性.....	162	6.4 I/O 子系统核心.....	209
5.1.2 文件操作.....	163	6.4.1 I/O 调度.....	209
5.1.3 文件类型.....	165	6.4.2 缓冲技术.....	209
5.1.4 文件结构.....	166	6.4.3 中断技术.....	210
5.2 文件访问方式.....	168	6.4.4 出错处理.....	211
5.2.1 顺序访问.....	168	6.4.5 I/O 请求处理.....	212
5.2.2 直接访问.....	168	6.5 磁盘结构.....	213
5.2.3 其他文件访问形式.....	169	6.6 磁盘调度.....	214
5.3 目录结构.....	170	6.6.1 FCFS 调度.....	214
5.3.1 一级目录.....	171	6.6.2 SSTF 调度.....	215
5.3.2 两级目录.....	172	6.6.3 SCAN 调度.....	215
5.3.3 树形目录结构.....	173	6.6.4 C-SCAN 调度.....	216
5.3.4 非循环图目录结构.....	175	6.6.5 LOOK 调度.....	216
5.3.5 一般图形目录结构.....	177	6.6.6 磁盘调度算法的选择.....	217
5.4 文件保护.....	178	6.7 磁盘管理.....	218
5.4.1 文件访问类型.....	178	6.7.1 磁盘格式化.....	218
5.4.2 访问表和组.....	179	6.7.2 引导块.....	219
5.4.3 其他文件保护方式.....	180	6.7.3 坏块.....	219
5.5 文件系统结构.....	180	6.8 空间交换管理.....	220
5.6 文件存储空间分配与管理.....	181	6.8.1 交换空间的使用.....	221
5.6.1 连续分配.....	182	6.8.2 交换空间分配.....	221
5.6.2 链接分配.....	183	6.8.3 交换空间管理.....	221
5.6.3 索引分配.....	185	6.9 磁盘可靠性.....	222
5.6.4 分配算法性能.....	187	6.10 稳定存储.....	223
5.6.5 空闲空间管理.....	188	小结.....	224

综合练习六.....	225	8.2.1 集中式算法.....	261
一、选择题.....	225	8.2.2 分布式算法.....	263
二、填空题.....	226	8.2.3 令牌环算法.....	264
三、简答题.....	226	8.2.4 三种算法比较.....	265
四、应用题.....	227	8.3 死锁处理.....	265
第7章 分布式通信.....	228	8.3.1 死锁预防.....	266
7.1 背景知识.....	228	8.3.2 死锁检测.....	267
7.1.1 分布式系统的优点.....	228	8.4 选举算法.....	269
7.1.2 网络拓扑结构.....	229	8.4.1 欺负 (bully) 算法.....	270
7.1.3 网络类型.....	231	8.4.2 环算法.....	271
7.2 通信.....	233	8.5 分布式文件系统.....	271
7.2.1 命名和名字协议.....	234	8.5.1 概述.....	271
7.2.2 路由策略.....	234	8.5.2 文件和目录服务.....	273
7.2.3 连接策略.....	236	8.5.3 命名透明性.....	274
7.2.4 竞争.....	237	8.5.4 文件命名方法.....	274
7.3 通信协议.....	237	8.5.5 文件复制.....	275
7.3.1 OSI 参考模型.....	237	8.6 分布式缓存管理.....	276
7.3.2 TCP/IP 参考模型.....	240	8.6.1 缓存的位置.....	277
7.3.3 差错检测.....	241	8.6.2 缓存的一致性.....	277
7.3.4 通信示例.....	242	8.7 系统示例: NFS.....	279
7.4 套接字编程.....	242	8.7.1 NFS 结构.....	279
7.4.1 Java 套接字.....	243	8.7.2 NFS 协议.....	280
7.4.2 TCP 套接字实例.....	244	8.7.3 NFS 实现.....	281
7.5 远程方法调用.....	246	8.8 分布式文件系统的发展.....	283
7.5.1 服务器开发.....	247	8.8.1 硬件.....	283
7.5.2 客户机开发.....	250	8.8.2 可扩充性.....	284
7.5.3 程序运行.....	254	8.8.3 广域网.....	284
7.6 CORBA.....	254	8.8.4 其他.....	284
7.6.1 服务器开发.....	255	小结.....	285
7.6.2 客户端开发.....	256	综合练习八.....	285
小结.....	256	一、选择题.....	285
综合练习七.....	257	二、填空题.....	286
一、选择题.....	257	三、简答题.....	287
二、填空题.....	257	四、应用题.....	287
三、简答题.....	258	第9章 系统安全问题.....	288
四、应用题.....	258	9.1 安全威胁.....	288
第8章 分布式管理.....	259	9.1.1 威胁的类型.....	288
8.1 事件排序.....	259	9.1.2 对计算机系统资源的威胁.....	289
8.1.1 前于关系.....	259	9.2 数据加密技术.....	291
8.1.2 实现.....	260	9.2.1 数据加密技术的发展.....	291
8.2 互斥现象.....	261	9.2.2 数据加密模型.....	292
		9.2.3 加密算法的类型.....	292

9.3 访问控制技术和认证技术.....	293	10.6.5 文件的存储.....	327
9.3.1 访问矩阵 (Access Matrix)	293	10.7 I/O 系统	329
9.3.2 认证 (authentication)	294	10.7.1 UNIX I/O 设备管理结构.....	330
9.4 入侵者.....	295	10.7.2 设备状态及设备控制.....	330
9.4.1 入侵技术.....	295	10.7.3 设备的读写方法	330
9.4.2 口令.....	296	10.8 网络.....	331
9.4.3 口令的选择.....	298	10.8.1 UNIX 网络分层结构.....	331
9.4.4 入侵检测.....	299	10.8.2 UNIX 网络协议	331
9.5 恶意软件	301	小结.....	332
9.5.1 恶意程序.....	301	综合练习十.....	332
9.5.2 病毒的本质.....	303	一、选择题.....	332
9.5.3 病毒的类型.....	303	二、填空题.....	333
9.5.4 宏病毒.....	304	三、简答题.....	333
9.5.5 反病毒方法.....	305	四、应用题.....	334
9.6 可靠系统	305	第 11 章 Linux 操作系统.....	335
小结.....	306	11.1 Linux 的历史.....	335
综合练习九.....	307	11.2 Linux 系统概述.....	336
一、选择题.....	307	11.2.1 Linux 内核.....	336
二、填空题.....	307	11.2.2 Linux Shell.....	336
三、简答题.....	308	11.2.3 Linux 文件结构.....	336
四、应用题.....	308	11.2.4 Linux 实用工具.....	337
第 10 章 UNIX 操作系统.....	309	11.3 Linux 内核概况.....	337
10.1 UNIX 的历史.....	309	11.4 进程管理.....	338
10.2 设计总则	311	11.4.1 进程和线程的基本概念.....	338
10.3 Shell 命令.....	314	11.4.2 如何管理进程.....	340
10.3.1 Shell 的定义和基本功能.....	314	11.4.3 进程调度.....	341
10.3.2 UNIX 系统中标准流的定义.....	314	11.4.4 Linux 运行模式.....	343
10.3.3 Shell 命令解释功能.....	314	11.4.5 内存管理.....	344
10.3.4 Shell 的过滤功能.....	316	11.5 文件系统.....	346
10.4 进程管理	317	11.5.1 Linux 核心文件系统简介	347
10.4.1 UNIX 中进程的描述.....	317	11.5.2 虚拟文件系统 (VFS)	348
10.4.2 进程的创建和消亡	318	11.5.3 EXTeNded 2 文件 系统 (EXT2fs)	349
10.4.3 UNIX 进程调度与管理.....	319	11.6 跨进程通信	350
10.4.4 UNIX 中进程的通信.....	320	11.6.1 信号.....	350
10.5 内存管理	321	11.6.2 管道.....	350
10.5.1 交换 (Swapping)	321	11.6.3 系统 V IPC 机制	351
10.5.2 分页 (Paging)	322	11.7 网络结构.....	351
10.6 文件系统	323	11.8 Linux 内核解读入门 (补充)	351
10.6.1 UNIX 中的文件.....	323	小结.....	352
10.6.2 文件系统设计	325	综合练习十一	353
10.6.3 文件的检索	326	一、选择题.....	353
10.6.4 文件 I/O	327		

二、填空题.....	353	12.8.4 网络服务函数.....	387
三、简答题.....	354	12.8.5 系统服务函数.....	387
四、应用题.....	354	小结.....	388
第 12 章 Windows 2003 操作系统.....	355	综合练习十二.....	388
12.1 Windows 2003 的历史.....	355	一、选择题.....	388
12.2 设计总则.....	356	二、填空题.....	389
12.3 系统架构.....	357	三、简答题.....	389
12.3.1 硬件抽象层 (HAL).....	358	四、应用题.....	389
12.3.2 内核 (Kernel).....	358	附录 A Java 语言基础.....	390
12.4 执行体.....	362	A.1 Java 语言简介.....	390
12.4.1 面向对象管理机制.....	362	A.1.1 一个 Java Application 程序.....	390
12.4.2 对象命名.....	362	A.1.2 数据类型.....	391
12.4.3 虚拟内存管理器.....	362	A.1.3 标识符.....	391
12.4.4 进程管理器.....	365	A.1.4 运算符.....	392
12.4.5 本地过程调用 (LPC).....	366	A.1.5 语句.....	392
12.4.6 I/O 管理器 (I/O 系统).....	367	A.1.6 函数.....	394
12.4.7 安全监控.....	367	A.1.7 类和对象.....	394
12.5 外部环境子系统.....	367	A.1.8 数组.....	396
12.5.1 Win32 子系统.....	368	A.1.9 包.....	397
12.5.2 POSIX 子系统.....	368	A.2 继承.....	397
12.5.3 OS/2 子系统.....	368	A.3 抽象类和接口.....	398
12.5.4 安全子系统.....	369	A.4 异常处理.....	402
12.6 Windows 2003 文件系统 (NTFS).....	369	A.5 Application 和 Applet.....	403
12.6.1 NTFS 在磁盘上的结构.....	369	参考答案.....	405
12.6.2 可恢复支持.....	373	第 1 章.....	405
12.6.3 NTFS 的安全性.....	374	第 2 章.....	406
12.6.4 容错支持.....	376	第 3 章.....	407
12.7 Windows 2003 网络架构.....	377	第 4 章.....	409
12.7.1 ISO/OSI 参考模型.....	377	第 5 章.....	411
12.7.2 传输协议.....	379	第 6 章.....	412
12.7.3 分布式应用机制.....	380	第 7 章.....	414
12.7.4 复位向与服务.....	381	第 8 章.....	415
12.7.5 域的基本管理.....	382	第 9 章.....	417
12.7.6 TCP/IP 网络命名机制.....	382	第 10 章.....	418
12.7.7 IIS (Internet 信息服务).....	384	第 11 章.....	419
12.8 编程接口.....	385	第 12 章.....	420
12.8.1 窗口管理函数.....	385	参考文献.....	422
12.8.2 国际性特性函数.....	386		
12.8.3 图形设备接口函数.....	386		

第 1 章 计算机操作系统概述

操作系统是一个作用于用户与计算机以及计算机硬件中的中介。操作系统的目的是为用户在执行程序的时候提供一个有效而且便利的环境。

可以从最早的手动操作系统到现在的多道程序和分时操作系统去追踪操作系统的发展。在了解这些发展的基础上，理解操作系统是做什么的和怎么去工作的。

操作系统必须保证计算机系统的操作正确性。为了阻止用户程序错误干预系统的操作，计算机硬件必须提供一个适当的机制去确保这些操作是正确的。

1.1 操作系统简介

人类使用计算机已经有 50 多年的历史了，但是计算机操作系统的历史并没有那么久，这是因为一开始计算机里并没有操作系统。在计算机出现 10 年左右时，人们逐渐觉得一台没有操作系统的计算机使用起来非常不方便，于是有科学家提出为计算机制造一个“管家”的概念，而这个“管家”就是操作系统。计算机操作系统经过几十年的发展，越来越符合人们高效使用计算机的要求。

通过了解计算机操作系统的发展历史，可以更好地了解什么是计算机操作系统，因此，在这一章里，主要讲述操作系统的发展，由此对操作系统有一个大体了解。

操作系统是系统程序。应用程序和计算机硬件的桥梁，因此操作系统承担着和硬件良好地接合的任务，尤其是和 I/O 设备的相互交流，另外，控制数据存储设备对数据进行操作也是操作系统的重要内容之一，因此，本章将先介绍基本的计算机结构，如 CPU、内存等，然后集中讨论 I/O 设备的访问和存储设备的访问。

1.1.1 什么是操作系统

电脑是由硬件和软件组成的，缺少任何一样都无法运行。对电脑进行操作，就是利用操作系统来完成的。最初的电脑没有操作系统，人们通过各种操作按钮来对其进行控制，后来出现了汇编语言，并将它的编译器内置到电脑中，操作人员通过有孔的纸带将程序输入电脑进行编译。这些将语言内置的电脑只能由操作人员自己编写程序来运行，不利于设备、程序的共用。为解决这种问题，出现了操作系统，这样就很好的实现了程序的共用，以及对计算机硬件资源的管理，使人们可以从更高层次对电脑进行操作，而不用关心其底层的运作。

操作系统是现代电脑必不可少的系统软件，是电脑的灵魂所在。现代的电脑都是通过操作系统来解释人们的命令，从而达到控制电脑的目的。几乎所有的应用程序都是基于操作系统的。

计算机硬件通常是由中央处理器、存储器、输入输出设备等部件组成，它们构成了系统本身和用户作业赖以活动的物质基础和工作系统。

计算机软件包括系统软件和应用软件。系统软件如操作系统、多种语言处理程序（汇编和编译程序等）、连接装配程序、系统实用程序、多种工具软件等；应用软件是为多种应

用目的而编制的程序。

没有任何软件支持的计算机称为裸机 (Bare Machine)，它仅仅构成了计算机系统的物质基础，而实际呈现在用户面前的计算机系统是经过若干层软件改造过的计算机系统。

接口示意图如图 1-1 所示。

层次模型如图 1-2 所示。

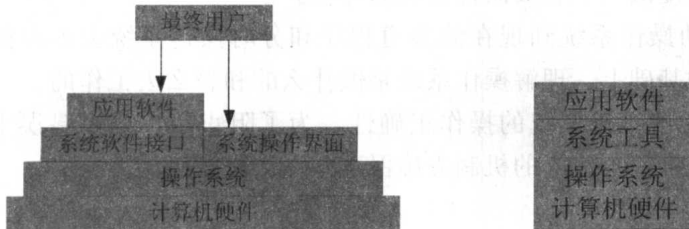


图 1-1 计算机系统的接口示意图

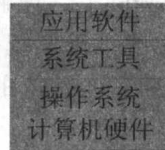


图 1-2 计算机系统的层次模型

给操作系统下定义很困难，至今没有一个统一的说法。以下只列举了现今操作系统教材中常见的几种从不同的观察角度定义操作系统的观点。

(1) 从自顶向下的角度看，操作系统是对裸机的第一层软件，是对机器的第一次扩展，为用户提供了与一台与实际硬件等价的虚拟机。

(2) 从自底向上的角度看，操作系统是资源管理在相互竞争的程序之间有序地对处理器、存储器以及其他 I/O 接口设备控制和分配。

(3) 从软件分类角度看，操作系统是最基本的系统软件，它控制着计算机所有的资源并提供应用程序开发的接口。

(4) 从系统管理员角度看，操作系统合理地组织管理了计算机系统的工作流程，使之能为多个用户提供安全高效的计算机资源共享。

(5) 从程序员角度看（即从操作系统产生的角度看），操作系统是将程序员从复杂的硬件控制中解脱出来，并为软件开发者提供了一个虚拟机，从而能更方便的进行程序设计。

(6) 从一般用户角度看，操作系统为用户提供了一个良好的交互界面，使用户不必了解有关硬件和系统软件的细节，就能方便地使用计算机。

(7) 从硬件设计者看，操作系统为计算机系统功能的扩展提供了支撑平台，使硬件系统与应用软件产生了相对独立性，可以在一定范围内对硬件模块进行升级和添加新硬件，而不会影响原来的应用软件。

总的来讲，传统的操作系统定义如下：

操作系统是控制和管理计算机系统内各种硬件和软件资源，合理有效地组织计算机系统的工作，为用户提供一个使用方便并且可扩展的工作环境，从而起到连接计算机和用户的接口作用。

早期计算机系统是用于科学与工程数值计算，如第一台计算机（1944 年）产生的原因就是解决大口径火炮设计中的计算问题：弹道轨迹、弹着点和材料的各种应力分布的计算以及原子弹设计和实验中大量的计算问题。

20 世纪 40 年代中期到 20 世纪 50 年代末，计算机处于电子管时代，没有操作系统，人们把这个时期称为“手工操作阶段”。这时的计算机大量需要人工控制，还没有“管家”来为它们服务。20 世纪 50 年代末，计算机的运行速度有了很大的提高，从每秒几千次、

几万次发展到每秒几十万次、上百万次，出现了联机批处理系统、脱机批处理系统、执行系统（执行系统不仅负责调度作业自动地运行，而且还要控制 I/O）。

20 世纪 50 年代末~20 世纪 60 年代中期，计算机进入了第二代——晶体管时代。为了解决人机矛盾，提高自动化程度，人们研制了监督程序，由该程序自动依次处理一系列任务。

20 世纪 60 年代中期~20 世纪 70 年代中期，计算机进入第三代——集成电路时代。在这一时期操作系统初步形成并完善，出现了三种最基本的操作系统类型：多道批处理操作系统、分时操作系统和实时操作系统。

20 世纪 80 年代至今，第四代计算机，大规模集成电路工艺技术飞速发展。操作系统也有了进一步发展：出现了个人计算机操作系统、网络操作系统和分布式操作系统。

1.1.2 批处理操作系统

在计算机发展的早期阶段，系统是让用户独占使用的，即在其使用期间，用户可以建立、运行它的作业，并最后作结尾处理。由于当时软件发展处于初期阶段，用于管理的软件还没有产生，因此所有的运行管理和具体操作都由用户自己承担。引入批量监督程序是为了实现作业建立和作业过度的自动化。监督程序是一个常驻内存的核心代码，每一种语言翻译程序（汇编语言或某种高级语言的编译程序）或实用程序（如链接程序）都作为监督程序的子例程。

1. 单道批处理系统

1) 单道批处理系统（Simple Batch Processing System）的处理过程

早期的计算机系统非常昂贵，为了能充分地利用它，应尽量让系统连续运行，以减少空闲时间，提高其利用率。为此，通常是把一批作业以脱机的方式输入到磁带上，并在系统中配上监督程序（Monitor），在监督程序的控制下使这批作业能一个接一个地连续处理。其自动处理过程是：首先，由监督程序将磁带（盘）上的第一个作业装入内存，并把运行控制权交给该作业。当该作业处理完成时，又把控制权交还给监督程序，再由监督程序把磁带（盘）上的第二个作业调入内存。计算机系统就是这样自动地一个作业一个作业地进行处理，直至磁带（盘）上的所有作业全部完成。由于系统对作业的处理都是成批进行的，而且在内存中始终只保持一道作业，故称为单道批处理系统。单道批处理系统的处理流程如图 1-3 所示。

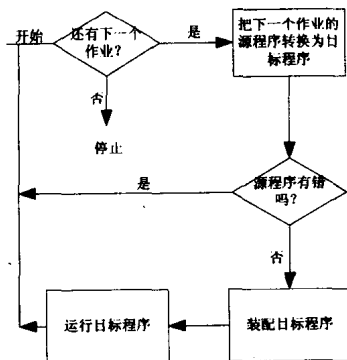


图 1-3 单道批处理系统的处理流程

由上述不难看出,单道批处理系统是在解决人机矛盾和 CPU 与 I/O 设备速度不匹配的矛盾的过程中形成的。换言之,单道批处理系统旨在提高系统资源的利用率和系统吞吐量。但这种单道批处理系统仍然不能很好地利用系统资源,故现在已经很少使用。

2) 单道批处理系统的特征

单道批处理系统是最早出现的一种计算机操作系统,但是在严格意义上来说,它只能算是计算机系统的前身而并非是目前人们所理解的操作系统。尽管如此,该系统比起人工操作方式已经有很大的进步。该系统的主要特征如下:

(1) 自动性。在顺利的情况下,在磁带上的一批作业能自动地依次执行,而无须人工干预。

(2) 顺序性。磁带上的各道作业是顺序地进入内存,各道作业的完成顺序与它们进入内存的顺序在正常情况下完全相同,亦即符合 FIFO (First in, First out)。

(3) 单道性。内存中仅有一道程序运行,即监督程序每次从磁带上只调入一道作业进入内存运行,当该作业完成或是发生异常时,才换入其后继作业进入内存。

2. 多道批处理系统

1) 多道程序设计技术

多道程序设计技术是在计算机上内存中同时存放几道相互独立的程序,使它们在管理程序的控制下,相互穿插地运行。当某道程序因某种原因不能继续运行下去时(如等待外部设备传输数据),管理程序便将另一道程序投入运行,这样可以使 CPU 以及各外部设备尽量处于忙碌状态,从而大大提高了计算机的使用效率。

多道程序设计技术使得几道程序在系统内并行工作。但在 Von Neumann 型计算机结构中(在单处理机的情况下),CPU 严格地按照指令计数器的内容顺序地执行每一个操作,即一个时刻只能有一个程序在处理机上执行。那么,如何理解多道程序的并行执行呢?多道程序设计技术可以实现同时被接受进入计算机的若干道程序相互穿插地运行,即当一个正在处理机上运行的程序要输入或输出而不能继续运行下去时,就把处理机让给另一道程序。所以,从微观上看,一个时刻只有一个程序在处理机上运行;从宏观上看,几道程序都处于执行状态,有的正在处理机上运行,有的正在打印结果,有的在输入数据,它们的工作都在向前推进。多道程序设计在单处理机上逻辑上的同时执行称为并发执行,它和多道程序同时在多个处理机上执行是有区别的。前者是逻辑上的并行,后者是物理上的并行。

2) 多道批处理系统的特征

在批处理系统中引入多道程序设计后,会使系统具有以下特征:

(1) 多道性。在内存中可同时驻有多道程序,并允许它们并发执行,从而有效地提高资源利用率和系统吞吐量。

(2) 无序性。多个作业完成的先后顺序与它们进入内存的顺序之间并无严格的对应关系,即先进入内存的作业可能较后完成甚至最后完成;而后进入内存的作业又可能先完成。

(3) 调度性。作业从提交给系统开始直至完成需要经过以下两次调度:首先是作业调度,这是按一定的作业调度算法,从外存的后备作业队列中选择若干个作业调入内存;其次是进程调度,指按一定的进程调度算法,从已在内存的作业中选择一个作业将处理机

分配给它，使之执行。

如图 1-4 所示为单道和多道程序运行情况。

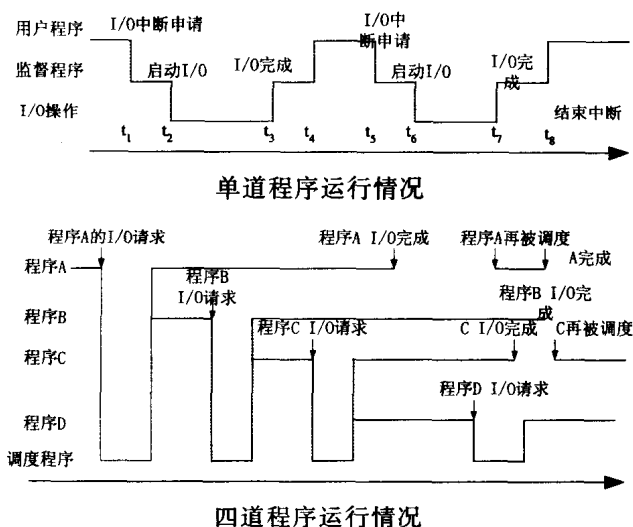


图 1-4 单道和多道程序运行情况

1.1.3 分时操作系统

在批处理方式下，用户以脱机操作方式使用计算机，用户在提交程序以后就完全脱离了自己的程序，在程序运行的过程中，不管出现什么样的情况都不能加以干预，只有等该批程序处理结束，用户才能得到计算结果。根据结果再做下一步处理，若有错，则重复上述过程，它的好处是计算机效率高。不过，用户仍十分留恋手工操作阶段的联机工作方式，可以独占计算机，并直接控制程序运行。但是独占计算机方式会造成资源效率低。因此既能保证计算机效率又能方便用户使用成为一种新的追求目标。

20 世纪 60 年代中期，计算机技术和软件技术的发展使这种追求成为了可能。由于 CPU 速度不断提高和采用分时技术，一台计算机可同时连接多个用户终端，每个用户可以在自己的终端上联机使用计算机，好像自己独占计算机一样。

所谓分时技术，就是把处理机的运行时间分成很短的时间片，按时间片轮流把处理机分配给各联机程序使用。若某个程序在分配给它的时间片内不能完成其计算，则该程序暂时中断，把处理机让给另一作业使用，等待下一轮时再继续其运行。由于计算机速度很快，作业运行轮转很快，给每个用户的印象就好像是每人独占一台计算机一样。每个用户可以通过自己终端向系统发出各种操作控制命令，完成作业的运行。

多用户分时操作系统是当今计算机操作系统中使用最普遍的一类操作系统。

1.1.4 实时操作系统

早期的计算机基本用于科学和工程问题的数值计算。上世纪 50 年代后期，计算机开始用于生产过程的控制，形成了实时系统。到了上世纪 60 年代中期计算机进入第三代，机器性能得到了极大的提高，整个计算机系统的功能大大增强了，使得计算机的应用领域越来越宽广。例如，炼钢、化工生产的过程控制，航天和军事防空系统中的实时控制。更为

重要的是计算机广泛用于信息管理,如仓库管理、医疗诊断、教学、气象、地质勘探直到图书检索、飞机订票、银行储蓄、出版编辑等。

实时操作系统是操作系统的又一类型。对外部输入的信息,实时操作系统能够在规定的时间内处理完毕并作出反应。“实时”二字的含义是指计算机对于外来信息能够及时进行处理并在被控对象允许的时间范围内作出快速反应。实时系统对响应时间的要求比分时系统更高,一般要求秒级、毫秒级甚至微秒级的响应时间。

电子计算机应用到实时控制时配置上实时操作系统,组成各种各样的实时系统。实时系统按其方式不同可以分为两类:

1. 实时控制系统

微型计算机的最早应用之一是进行过程控制和提供环境监督。过程控制系统是从传感器获得输入的数字或模拟信息进行分析处理后,激发一个改变可控过程的活动信号,以达到控制的目的。

2. 实时信息处理系统

电子计算机还有一类很重要的实时性应用是组成实时数据处理系统,比如自动订票系统、情报检索系统等。这一类应用大多数用于服务性行业,用户可以通过终端设备向计算机提出某种要求,而计算机系统处理后将通过终端设备回答用户。

1.1.5 分布式操作系统

一组相互连接并能交换信息的计算机形成了一个网络。这些计算机之间可以相互通信,任何一台计算机上的用户都可以共享网络上其他计算机的资源。但是,计算机网络并不是一个一体化的系统,它没有标准的、统一的接口。网络上各站点的计算机有各自的系统调用命令、数据格式等。若一台计算机上的用户希望能使用网上的另一台计算机的资源,则必须指明是哪个站点上的哪台计算机,并以那台计算机上的命令、数据格式来请求才能实现共享。另外,为完成一个共同的计算机任务,分布在不同主机上的各交互进程的同步协作也难以实现。因此计算机网络的功能对用户来讲是不透明的。它存在两个问题:一是在网络上不同的类型计算机中,对某一种计算机所编写的程序无法在另一类计算机上运行;二是不能在具有不同数据格式、字符编码的计算机之间实现数据共享。另外,还需要解决分布在不同主机上的诸交互进程无法自动实现紧密合作的问题。

大量的实际应用要求一个完整的一体化系统,而且又具有分布处理能力。如在分布事务处理、分布数据处理、办公自动化系统等实际应用中,用户希望以统一的界面、标准的接口去使用系统的各类资源,去实现所需要的各种操作。这就导致了分布式系统的出现。

一个分布式系统就是若干台计算机的集合,这些计算机都有自己的局部存储器和外部设备,它们既可以独立工作(自治性),也可以合作。在这个系统中各机器可以并行操作而且有多个控制中心,即具有并行处理和分布控制的功能。分布式系统是一个一体化的系统,在整个系统中有一个全局的操作系统称为分布式操作系统,它负责全系统的资源分配和调度、任务划分、信息传输、控制协调等工作,并为用户提供统一的界面、标准的接口。用户通过这一界面实现所需要的操作和使用系统的资源。至于操作是在哪一台计算机上执行或是使用了哪台计算机的资源则是系统的事,用户不用知道,也就是说系统对用户是透明的。