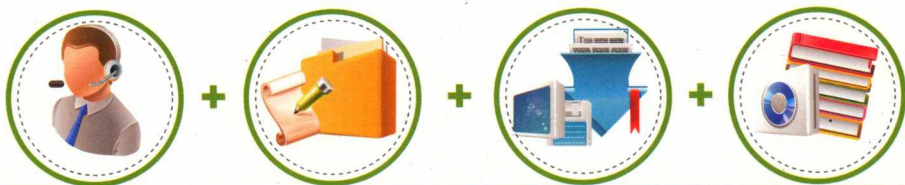


操作系统

原理及应用

主 编 陈 敏
副主编 许雪林 汤龙梅
王 璇 杨海燕



- ◆ 以基础知识—实用技术—项目实训为主线
- ◆ 按照教与学的实际需要取材谋篇
- ◆ 精心设置“小型案例实训”和实验，旨在培养学生的实践能力
- ◆ 配备免费教学资源——电子课件、习题答案等



全国高等院校应用型创新规划教材·计算机系列

操作系统原理及应用

陈 敏 主 编

许雪林 汤龙梅

副主编

王 璇 杨海燕

清华大学出版社
北京

内 容 简 介

操作系统是现代计算机中必不可少的核心软件，是计算机学科各专业的必修课程，也是从事计算机应用人员的必备知识。

本书系统地讲述了操作系统的基本概念、原理、技术、策略及功能，尽可能多方面地提示操作系统的精髓和特征，以简洁、易懂的语言组织全书内容。

全书共7章，第1章介绍操作系统的基本知识，第2章详细说明进程管理的相关内容，第3章阐述处理机调度，第4章介绍存储管理，第5章介绍设备管理，第6章介绍文件系统，第7章介绍Linux网络及服务器配置实例。

本书既可作为高等院校计算机及相关专业本科、专科的教材，也可供从事计算机科学、工程、应用等方面工作的科技人员参考使用。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

操作系统原理及应用/陈敏主编. —北京：清华大学出版社，2017

(全国高等院校应用型创新规划教材·计算机系列)

ISBN 978-7-302-47892-8

I. ①操… II. ①陈… III. ①Linux 操作系统—高等学校—教材 IV. ①TP316.85

中国版本图书馆CIP数据核字(2017)第193377号

责任编辑：汤涌涛

封面设计：杨玉兰

责任校对：李玉茹

责任印制：王静怡

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦A座 邮 编：100084

社总机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载：<http://www.tup.com.cn>, 010-62791865

印 装 者：北京密云胶印厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：16.75 字 数：390千字

版 次：2017年10月第1版 印 次：2017年10月第1次印刷

印 数：1~2000

定 价：39.80元

产品编号：073377-01

前 言

操作系统是计算机系统中较为重要的系统软件，在计算机学科的课程体系中占有重要的地位，是计算机及相关专业的一门基础必修课，也是计算机专业从业者必须掌握的知识。一本适用的教材对于操作系统的学习尤为重要。因此，作者在多年教学工作的基础上，结合 Linux 2.4 内核相关内容编写了此书。

考虑到课程学习的有限课时数，我们对内容进行了精选，着重于操作系统基本概念、基本原理、实现策略、基本算法原理的阐述，力图从两个主线——操作系统的资源管理角度和面向用户的角度将操作系统内容组织成一个逻辑清晰的整体。

全书共分 7 章。从操作系统的资源管理角度分别介绍了相关软硬件资源管理的内容，并在其中引入 Linux 2.4 相关的内容进行实例说明。

第 1 章 概述 介绍操作系统的基本知识、操作系统的历史与发展、操作系统的分类，简要介绍了计算机系统相关部件，并引入系统调用的概念，说明了操作系统的特征及发展趋势，并对 Linux 操作系统的产生及发展特征做了简要说明。

第 2 章 进程控制 介绍进程的概念，对进程控制、进程互斥、同步、通信、进程死锁、管程、线程的概念等问题进行了分析和讨论，并介绍了 Linux 进程控制、Linux 进程通信的内容，设计了两次实验。

第 3 章 处理机调度 介绍作业的概念、作业与进程的关系、多级调度的概念、作业及进程调度算法、Linux 进程调度等相关内容。

第 4 章 存储管理 介绍存储管理功能、单一连续存储管理、分区式管理、分页式管理、分段式管理、段页式管理、虚拟存储技术、Linux 存储管理等知识。

第 5 章 设备管理 介绍设备管理概述、设备控制器、设备的数据传输控制方式、中断技术、缓冲技术、设备独立性、设备分配、SPOOLing 技术等内容。

第 6 章 文件系统 介绍文件的基本概念、文件组织形式、文件存储空间管理方法、文件目录管理、文件操作、文件系统的层次模型、Linux 文件系统概述等内容。

第 7 章 Linux 网络及服务器配置实例 介绍 Linux 网络基础知识、网卡配置、Linux 网络服务、samba 服务器配置、DNS 服务器配置、FTP 服务器配置等相关内容。

本书主要由福建工程学院陈敏、许雪林、汤龙梅、王璇、杨海燕等教师合作编写，在本书的编写过程中参考了大量的相关技术资料及经典案例，吸取了许多宝贵经验，在此一并表示谢意！

由于编者水平有限，书中难免会有疏漏和不妥之处，希望读者批评指正。作者 E-mail: chenmin@fjut.edu.cn。

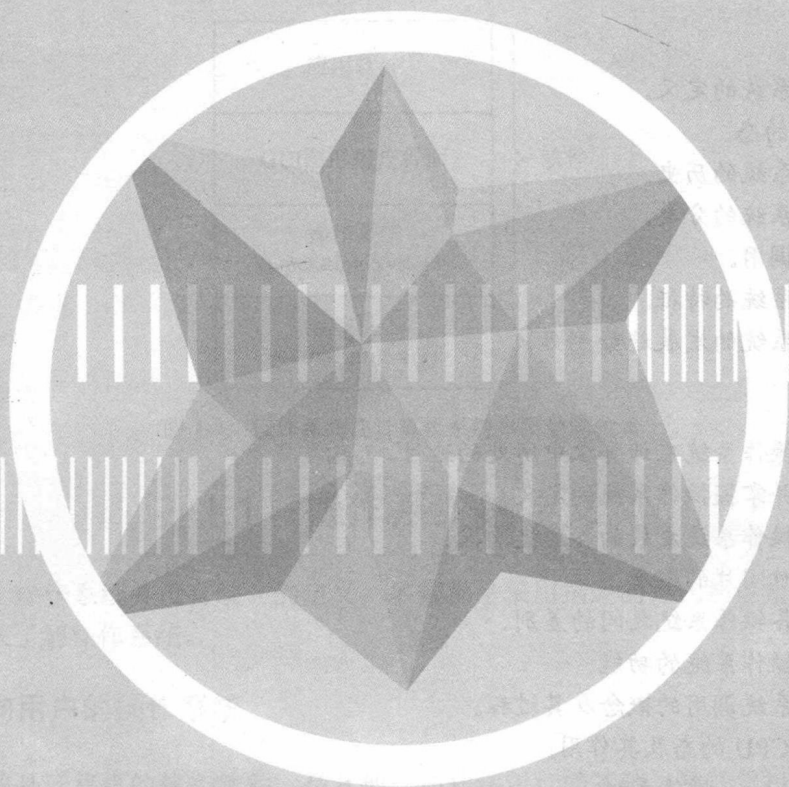
编 者

第 1 章 概述1	2.1.1 程序的顺序执行..... 27
1.1 操作系统的定义.....3	2.1.2 程序的并发执行..... 28
1.1.1 面向用户的操作系统.....3	2.1.3 进程的引入原因..... 31
1.1.2 面向系统资源的操作系统.....4	2.2 进程控制 38
1.1.3 操作系统的定义.....4	2.2.1 进程创建..... 38
1.2 操作系统的发展简史.....5	2.2.2 进程撤销..... 40
1.2.1 手工操作阶段.....5	2.2.3 进程阻塞..... 41
1.2.2 监督程序阶段.....5	2.2.4 进程唤醒..... 42
1.2.3 执行系统阶段.....5	2.2.5 Linux 进程创建及执行实例 43
1.2.4 多道程序系统阶段.....6	2.3 进程间通信 45
1.2.5 操作系统的形成.....7	2.3.1 临界区与临界资源..... 46
1.3 操作系统的分类.....7	2.3.2 忙等的互斥..... 46
1.4 计算机系统硬件简介.....11	2.3.3 用信号量机制实现互斥与 同步..... 50
1.4.1 中央处理器.....11	2.3.4 进程通信..... 56
1.4.2 存储器.....11	2.3.5 Linux IPC 概述..... 57
1.4.3 磁盘.....12	2.3.6 Linux 管道通信..... 58
1.4.4 I/O 设备.....12	2.3.7 Linux 软中断通信..... 60
1.4.5 总线.....13	2.4 经典的 IPC 问题 63
1.4.6 操作系统的工作过程.....13	2.4.1 生产者与消费者问题..... 63
1.5 系统调用.....15	2.4.2 哲学家进餐问题..... 63
1.5.1 系统调用的基本类型.....16	2.4.3 读者-写者问题..... 65
1.5.2 系统调用的实现.....16	2.5 线程 66
1.6 现代操作系统的特征.....17	2.5.1 线程的引入及定义..... 66
1.7 操作系统的发展趋势.....19	2.5.2 线程与进程的关系..... 67
1.8 Linux 操作系统简介.....20	2.5.3 线程的实现方法..... 68
1.8.1 Linux 的产生.....20	2.5.4 Linux 的线程管理..... 68
1.8.2 Linux 的特性.....20	2.5.5 Linux 线程管理相关函数..... 69
1.8.3 Linux 与 Windows 操作系统 之间的差别.....21	2.5.6 Linux 线程管理举例..... 70
1.8.4 Linux 的用户界面.....22	2.6 管程 71
本章小结.....23	2.6.1 管程的提出..... 71
习题.....24	2.6.2 管程概念..... 71
第 2 章 进程控制25	2.6.3 管程的组成..... 71
2.1 进程的概念.....27	2.6.4 管程的形式..... 72

2.6.5 管程的三个主要特性.....	73	4.3 分区存储管理.....	112
2.7 死锁.....	73	4.3.1 固定分区方式.....	112
2.7.1 死锁的定义和起因.....	73	4.3.2 可变分区方式.....	114
2.7.2 规避死锁的方法.....	74	4.3.3 分区式存储管理的特点.....	116
2.8 小型案例实训.....	77	4.3.4 分区式存储管理的内存扩充 技术.....	117
本章小结.....	78	4.4 分页存储管理.....	119
习题.....	78	4.4.1 分页存储管理的基本原理.....	119
第3章 处理机调度	81	4.4.2 分页存储管理的数据结构.....	120
3.1 作业的概念.....	82	4.4.3 页式地址结构及转换.....	122
3.2 作业与进程的关系.....	83	4.4.4 相联存储器和快表.....	124
3.3 多级调度的概念.....	84	4.4.5 分页存储管理的内存分配与 回收.....	125
3.3.1 高级调度.....	84	4.4.6 分页存储管理的内存共享与 保护.....	126
3.3.2 中级调度.....	85	4.4.7 分页存储管理的特点.....	127
3.3.3 低级调度.....	85	4.5 分段存储管理.....	127
3.3.4 线程调度.....	85	4.5.1 分段存储管理的基本原理.....	127
3.4 调度算法.....	85	4.5.2 分段存储管理的数据结构.....	128
3.4.1 作业调度算法.....	86	4.5.3 分段存储管理的地址结构及 转换.....	128
3.4.2 进程调度算法.....	87	4.5.4 内存分配与回收.....	129
3.5 进程调度的时机.....	91	4.5.5 段共享与保护.....	130
3.6 Linux 进程调度.....	91	4.5.6 分段存储管理的特点.....	131
3.6.1 Linux 进程调度的目标.....	91	4.5.7 分页存储管理和分段存储管理 的区别.....	131
3.6.2 Linux 进程分类.....	92	4.6 段分页存储管理.....	131
3.6.3 Linux 进程优先级.....	92	4.7 虚拟存储管理系统.....	131
3.6.4 Linux 进程调度程序.....	93	4.7.1 虚拟存储概述.....	131
3.6.5 进程调度策略.....	94	4.7.2 请求分页虚拟存储管理.....	133
3.6.6 可变优先级.....	94	4.7.3 请求分段虚拟存储管理.....	138
3.6.7 可变时间片.....	94	4.7.4 段页式虚拟存储管理.....	138
3.6.8 Linux 进程调度实现.....	95	4.8 Linux 的存储管理.....	139
3.7 小型案例实训.....	95	4.8.1 Intel 80386 体系结构下的内存管 理机制.....	139
本章小结.....	103	4.8.2 Linux 地址转换机制.....	142
习题.....	103		
第4章 存储管理	106		
4.1 存储管理概述.....	107		
4.1.1 计算机系统的存储体系.....	108		
4.1.2 存储器的组织方式.....	109		
4.2 单一连续存储管理.....	112		

4.8.3 Linux 内存分配和释放	143	5.10 小型案例实训	178
4.8.4 Linux 进程虚拟内存地址	145	本章小结	182
4.8.5 Linux 页面操作	147	习题	183
4.9 小型案例实训	149	第 6 章 文件系统	184
本章小结	153	6.1 文件	186
习题	153	6.1.1 文件管理的几个基本概念	186
第 5 章 设备管理	155	6.1.2 文件分类	187
5.1 概述	156	6.2 文件组织形式	187
5.1.1 设备的分类	156	6.2.1 文件的逻辑结构	187
5.1.2 设备管理的功能	157	6.2.2 文件的物理结构	188
5.2 设备控制器	158	6.3 文件存储空间的管理	191
5.2.1 设备控制器的基本功能	158	6.3.1 空闲文件目录法	191
5.2.2 设备控制器的组成	159	6.3.2 空闲块链法	191
5.3 数据传送控制方式	159	6.4 文件目录管理	192
5.3.1 程序直接控制方式	160	6.4.1 文件目录管理的基本要求	192
5.3.2 中断控制方式	160	6.4.2 文件控制块和索引节点	193
5.3.3 DMA 方式	161	6.4.3 文件目录结构	193
5.3.4 通道控制方式	163	6.4.4 文件共享	194
5.4 中断技术	164	6.4.5 文件保护	195
5.4.1 中断的基本概念	164	6.5 文件操作	196
5.4.2 中断类型	166	6.6 文件系统的层次模型	196
5.4.3 中断的优先级	167	6.7 Linux 文件系统概述	198
5.4.4 软件中断	168	6.7.1 Linux 文件系统特点	198
5.5 缓冲技术	168	6.7.2 Linux 的文件类型	198
5.5.1 缓冲技术的引入	168	6.7.3 Linux 的虚拟文件系统	199
5.5.2 缓冲的分类与管理	169	6.7.4 挂载、卸载文件系统	201
5.6 设备独立性	169	6.7.5 ext2 文件系统	201
5.7 设备分配	170	6.8 小型案例实训	202
5.7.1 设备分配中的数据结构	170	本章小结	208
5.7.2 设备分配时应考虑的因素	171	习题	209
5.8 SPOOLing 系统	172	第 7 章 Linux 网络及服务器配置	
5.9 Linux 设备管理	173	实例	211
5.9.1 设备管理概述	173	7.1 Linux 网络基础	212
5.9.2 Linux 的 I/O 控制	175	7.1.1 Linux 网络的相关概念	212
5.9.3 字符设备与块设备管理	177		

7.1.2	Linux 的网络端口	212	7.4.7	编辑文件配置 samba 服务器 实例	230
7.1.3	Linux 网络的相关配置文件	213	7.5	DNS 服务器	232
7.2	配置网卡	216	7.5.1	配置主 DNS 服务器	232
7.2.1	配置 TCP/IP 网络	216	7.5.2	配置辅助 DNS 服务器	234
7.2.2	网络相关命令	217	7.5.3	测试 DNS 服务器	234
7.2.3	桌面环境下配置网卡	218	7.6	FTP 服务器	236
7.2.4	配置网络的 shell 命令	221	7.6.1	FTP 服务器简介	236
7.3	Linux 网络服务	223	7.6.2	vsftpd 服务器配置基础	237
7.3.1	服务器软件与网络服务	223	7.6.3	配置 vsftpd 服务器	239
7.3.2	管理服务	224	本章小结	240	
7.4	samba 服务器	224	习题	241	
7.4.1	samba 概述	224	附录	242	
7.4.2	samba 的安装	227	参考文献	257	
7.4.3	samba 的配置文件	227			
7.4.4	samba 的文件共享	227			
7.4.5	samba 的打印共享	229			
7.4.6	启动和停止 samba 服务	230			



第 1 章

概 述

本章要点

- 操作系统的定义。
- CPU 的态。
- 操作系统的历史。
- 操作系统的分类。
- 系统调用。
- 操作系统的特征。
- 操作系统的发展趋势。

学习目标

- 理解操作系统在计算机中所处的地位。
- 理解并掌握操作系统的定义。
- 了解操作系统发展的各历史阶段。
- 理解时间片的概念。
- 理解各操作系统之间的差别。
- 理解操作系统的功能。
- 理解系统调用的概念及其过程。
- 理解 CPU 的态及其作用。
- 了解 Linux 操作系统。
- 熟练掌握操作系统课程学习的两条主线。

计算机硬件由处理器、内存、硬盘、显示器、键盘、鼠标、各种标准接口及其他各类 I/O 设备构成。整个计算机是一个复杂的系统。计算机的使用者一般是普通用户和编程人员。一方面，对程序员来说，如果只有了解系统的所有细节才能编写出程序，这将会限制程序员的开发工作；另一方面，如果要求普通用户在使用计算机时了解计算机内部工作原理，将极大地限制用户的使用，从而限制计算机的推广应用。因此，计算机中需要有一种软件将内部的细节和工作原理对所有用户屏蔽，并提供给用户一个统一、简便的使用环境，这个软件就是操作系统。

由图 1-1 可以看出，计算机的硬件与软件之间是一种层次结构的关系。计算机的各种实用程序和应用程序都运行在操作系统之上，操作系统是介于计算机硬件与用户软件之间的一种系统软件。通常把未配置任何软件的计算机称为裸机，而一般用户难以直接使用裸机。操作系统是计算机硬件的扩充，裸机+操作系统=虚拟机。一个裸机在每加上一层软件后，就变成了功能更强的虚拟机。

操作系统虽然也是一种软件，但它与应用软件(也称用户软件)是有很大差别的。这个差别除了用系统软件和应用软件来区分操作系统和其他用户软件之外，还可以从操作系统的管理功能上来加以区分。操作系统在计算机中起着资源管理的功能，它负责系统中的硬件及软件资源的管理，因此操作系统在计算机中扮演着举足轻重的作用。

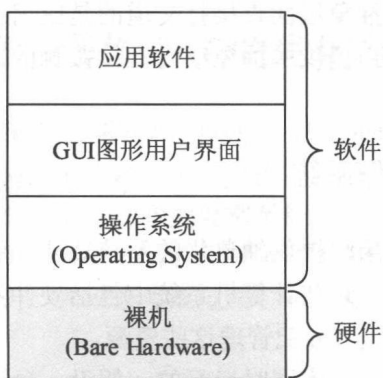


图 1-1 操作系统在计算机系统中所处的位置

1.1 操作系统的定义

目前对于操作系统并没有统一的定义，多数都是采用描述的方式加以定义。下面先从不同的角度来了解操作系统。

1.1.1 面向用户的操作系统

操作系统是计算机的核心软件，是其他一切软件运行的基础，是计算机开发的基础平台。操作系统在用户和计算机硬件之间架起了一个桥梁，通过这个桥梁，既可方便用户的使用，又可高效地发挥计算机硬件的功能，如图 1-2 所示。

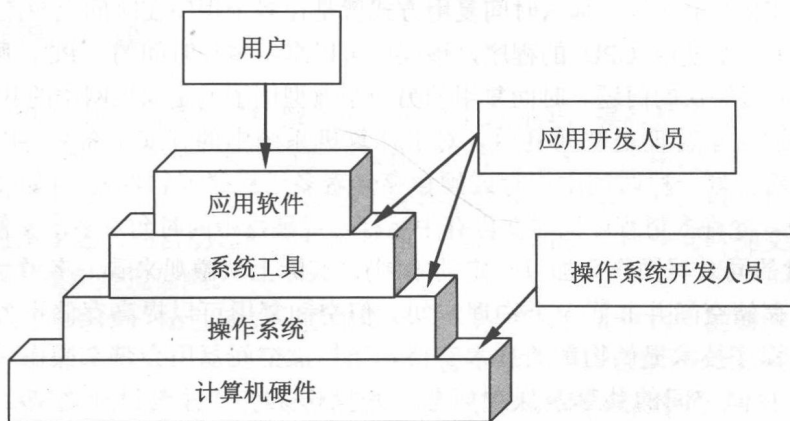


图 1-2 操作系统在计算机系统中的地位

操作系统在面向用户时，将内部工作细节对用户屏蔽，使用户不必关心开发设计的软件如何在系统中运行。同时，用户在使用计算机时不必关心软件资源如何存储、硬件资源何时可以使用等诸多细节问题。用户只需要关心程序是否被正确执行，提交的请求是否正确完成即可。这也就是将计算机系统的底层细节全部对用户抽象化。

注意：抽象其实是将复杂问题简化处理的一种方式。

实际上，与操作系统及其抽象层面直接打交道的是应用程序。而从另一方面来看，最终用户是与操作系统的用户接口提供的抽象层面直接接触的，而用户接口一般有图形界面接口、命令行接口等。

1.1.2 面向系统资源的操作系统

自顶向下看，操作系统为用户提供抽象化的系统功能。而自底向上看，操作系统主要负责管理系统的各个组成部分。现代计算机系统中包括硬件资源和软件资源，操作系统的作用就是在多任务和多用户环境下分配管理这些资源。

现代操作系统一般都是多道程序同时运行的。假设一台计算机上执行的多个程序碰巧在某一时刻同时提交打印请求到一台打印机上，打印时可能前一部分是某一程序的内容，接着打印另一程序的内容，然后又又是其他程序的内容，这种情况将会使计算机变得一团混乱。但是通过在硬盘上设置打印缓冲区，可解决同时提交打印请求的定序问题，以排除潜在的打印混乱问题。同时，其他程序可以继续提交打印请求，而显然这些程序的打印请求并未真正送到打印机。

当一台计算机同时存在多个用户时，系统中硬件和软件资源的管理变得更加复杂，因为这涉及资源的共享、分配及保护问题。在这种情况下，操作系统必须能够记录哪个用户在使用哪些资源，对资源的请求进行分配管理，计算资源用量，并能协调多个用户间的资源请求的冲突等问题。

对于系统资源的共享管理分为时间和空间上的复用。在时间上复用，就意味着某一资源在时间上供各程序轮流使用。例如，一般 PC 上只有单一 CPU，而多任务环境下会有多个程序需要在 CPU 上运行，那么时间复用方式就是让多个程序在时间上轮流使用 CPU。至于如何选择下一个使用 CPU 的程序，该程序可以享用多长时间的 CPU，则是操作系统在设计实现中应该解决的问题。时间复用的另一个典型例子就是局域网中的共享打印机。

另一个资源共享管理是空间复用。对于计算机系统存储设备——内存和硬盘来说，从宏观上看，同一段时间中内存或硬盘存储着多道程序或软件资源(如文件、数据库等)，也就是说，允许多道程序同时在内存中运行，或硬盘中同时允许多个文件的读写操作(当然对于硬盘的同时写操作会加以一定的限制)。实际上从微观来看，多道程序或多个软件资源占用的存储空间并非同一个物理空间。但空间复用可以提高存储设备的空间利用率，也为多道程序技术提供物理的技术支持。当然，空间复用会带来操作系统管理程序的复杂度，如存储空间的共享及保护问题，是操作系统软件在设计之初必须予以解决的。因此，操作系统对于存储空间的管理必须记录哪个程序占用哪部分存储空间，同时还需要记录存储空间的使用情况。

1.1.3 操作系统的定义

尽管操作系统尚无严格、统一的定义，但是一般认为操作系统的定义是：管理计算机系统软、硬件资源，控制程序执行，改善人机界面，并合理组织计算机工作流程和为用户提供简便使用环境的系统软件。

1.2 操作系统的发展简史

操作系统的发展伴随着计算机体系结构的发展。计算机的发展大致经历了如下几代。

第一代计算机(1945—1955): 电子管和手工操作。

第二代计算机(1955—1965): 晶体管和批处理系统。

第三代计算机(1965—1980): 集成电路芯片和多道程序设计技术。

第四代计算机(1980—1990): 大规模集成电路芯片和传统操作系统。

1.2.1 手工操作阶段

最初的计算机是没有操作系统的, 都是通过手工操作使用计算机。当作业需要在计算机上计算时, 先将事先准备好的程序和数据在卡片上穿孔, 或者将程序和数据存放在磁带上, 然后把穿孔卡片或磁带连接到读卡机、磁带机上, 将所需的器件手工连接好后, 就可以让计算机开始计算了, 计算完毕时, 再将计算结果打印或输出到磁带上。以上步骤完成后, 就可允许另一个用户使用计算机了。手工操作的特点是: 上机用户独占计算机全部资源, 手工操作时, 用户必须熟悉计算机各器件的所有细节, 因为若中间某一环节出错, 所有操作都必须重新开始。

在手工操作阶段, 程序员采用数字编码的机器语言进行编程, 随后产生的汇编语言将原来的数字编码替换为助记符。

随着计算机硬件的发展, 计算机的运算与人工操作之间速度匹配的矛盾越来越突出, 人们开始考虑尽可能地减少人工干预, 于是就出现了批处理。

1.2.2 监督程序阶段

20世纪50年代, 为减少系统操作员手工操作的时间, 提高系统资源的利用率, 人们开始利用计算机系统软件来代替系统操作员的部分工作。采用的办法是利用一个管理程序对重复的“装入→编译→执行→输出”操作过程实现自动控制, 可以识别和装入所需程序, 能处理作业之间的自动过渡和自动定序, 用户可将多个作业同时提交给系统处理, 这种程序处理方式就是批处理。

批处理系统的基本设计思想是: 操作员将若干用户的作业中相似的操作合并成一批, 如输入操作、输出操作、计算操作等, 在设备准备好之后, 启动一个常驻于内存的程序——监督程序(Monitor), 然后由监督程序自动控制完成这批作业的执行。

在监督程序的控制下实现作业的执行及作业之间的自动过渡, 可以缩短作业之间手工操作准备时间, 减少人工的干预和操作, 尽可能让计算机连续执行作业。

1.2.3 执行系统阶段

批处理实现了作业的自动过渡, 改善了计算机资源的使用情况。但是, 批处理系统仍存在不足之处, 如磁带需要人工装卸, 这样操作既麻烦又容易引发程序操作的不安全性。在批处理操作过程中, 没有一种监控程序, 可能会引起用户程序篡改监督程序等问题。在

批处理过程中，若程序出错，用户是无法对程序进行干预的，只能再次重新执行。

20 世纪 60 年代末，硬件获得两方面的进展，即通道技术和中断技术的出现，使操作系统进入了执行系统阶段。

通道是一种专用部件，它能控制多台外设同时工作，负责外设与内存间的信息传输。通道启动后能独立于 CPU 运行，即通道与 CPU 能并行工作。

中断是指当主机接到外部信号时(如需要进行外设操作)，暂停当前工作的执行，转去处理信号产生的事件，处理完毕再返回原来被中止的工作继续执行。

借助于通道和中断技术，输入输出工作可在主机控制下完成。这时，原有的监督程序功能已经扩大，它不仅负责调度作业的自动运行，还提供输入输出的控制功能。这个功能扩展的监督程序被称为执行系统。

1.2.4 多道程序系统阶段

执行系统克服了批处理系统的不足之处，但是作业的处理仍然是串行的，系统资源利用率不高。通道和中断技术的引进，使 CPU 和 I/O 设备，CPU 和通道的并行操作成为可能，这时多道程序的概念才变成现实。这里的“多道”，是指在内存中装入多个作业同时运行，使它们共享系统资源。

多道批处理系统(Multiple Batch Processing System)的基本思想是：将用户的作业先在外存排队，形成一个后备队列，由作业调度程序按照一定的策略从中选择若干作业进入内存，这些进入内存的作业共享系统中的各种资源，各作业在内存中交替使用处理器。

单道和多道程序系统的运行示意如图 1-3 和图 1-4 所示。

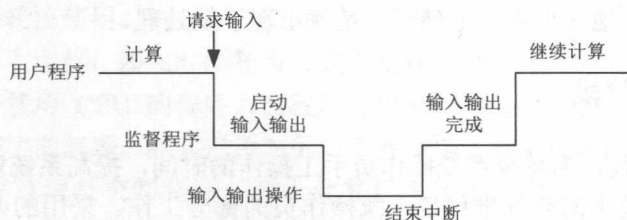


图 1-3 单道程序运行示意

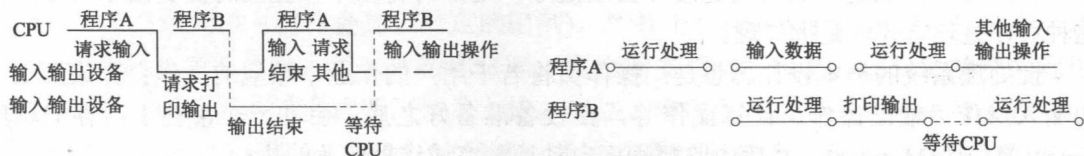


图 1-4 多道程序运行示意

多道批处理系统的特点如下。

- (1) 多道性。内存中同时有多个作业并发执行。
- (2) 宏观上并行。同时进入系统的几道程序都处于运行过程中，它们或于不同时刻开始，或于不同时刻结束，但在某段时间内总体上看是同时运行的。
- (3) 微观上串行。在内存中的多个作业的运行顺序与其进入内存的先后是没有关系的。
- (4) 多道技术会延长作业的周转时间。

引入多道程序设计技术,可以提高 CPU 的利用率,充分发挥计算机硬件的并行性。

多道程序系统(Multi Programming)与多重程序系统(Multi Processing)有必要加以区分。多重处理系统中配备有多个处理器,并配以相应的程序控制系统(即允许多个程序能并行执行),使多个程序同时运行。多重系统必须实现多道程序设计技术,这是物理硬件的软件条件支撑。但是,多道程序系统却不必一定有多重程序系统的支持。

实现多道程序设计技术必须解决以下 3 个问题。

(1) 存储保护与地址重定位。

在多道程序系统下,内存中通常装入多个程序,因此系统必须能够提供一种措施,让每道程序只能访问内存中自己的区域,以避免相互干扰。当某道程序发生错误时,不影响其他程序的执行,也不会影响系统程序,这就是存储保护。另外,为方便程序员编写程序,编写程序时无须知道程序在内存中的具体位置,因此当程序被执行时,应该由系统完成用户使用的地址与实际内存物理地址的转换,而当程序在内存中被移动时,也不会影响程序的执行,这就是地址重定位技术。

(2) 处理器的管理与分配。

系统中同时有多道程序可投入运行,但处理器只有一个,因此会存在将处理器分配给哪个程序的问题,即处理器分配调度的策略问题(详见第 3 章)。

(3) 资源的管理与调度。

系统中的资源包括硬件和软件资源,当涉及共享资源使用权的归属问题时,系统需按一定的策略来分配和调度,既要解决共享软硬件资源的竞争、共享及安全使用等问题,还要解决资源的利用率问题。

1.2.5 操作系统的形成

多道程序系统继承了批处理和执行系统的特点,它使作业操作过程更加自动化。随着计算机硬件技术的快速发展,机器运行速度进一步提高,存储容量也获得较大提升,设备数量和种类增多,这些都为软件发展提供了很好的支持。为了充分发挥硬件的功能,更好地满足用户的各类应用需求,管理程序必须满足更高的要求。

中断和通道技术使 CPU 与外设之间的并行方式工作成为可能,实现多道程序的技术条件已基本满足,但是,要真正实现系统管理程序的功能,就对大容量高速存储器提出了要求。20 世纪 60 年代磁盘的出现解决了这一需求问题,随之出现了多道批处理操作系统、分时操作系统和实时操作系统,这标志着操作系统正式形成。硬件系统中配置了操作系统之后,使管理程序的功能得到了进一步提高,可以完成处理器管理、存储器管理、设备管理、文件管理等功能。

1.3 操作系统的分类

操作系统发展的各个阶段使用不同的策略为用户提供不同的服务。为用户提供的处理方式不同,计算机的特征也就不一样,从而在用户面前呈现出具有不同处理方式和不同运行特点的操作系统。根据功能、特点和使用方式,可将操作系统分为以下几种类型。

1. 批处理操作系统

采用批处理方式工作的操作系统通常称为批处理操作系统(Batch Processing Operating System), 早期计算机中运行的 DOS 就是典型的批处理操作系统。批处理操作系统根据程序的执行方式分为单道批处理(也称为早期批处理)和多道批处理。

单道批处理系统(Simple Batch Processing System)是早期出现的一种系统。在系统中每次只有一个作业在运行, 当该作业运行完毕或出现错误时, 由监督程序控制自动装入下一作业继续运行, 这样就减少了作业切换时人工操作的时间, 实现了作业的运行过渡, 从一定程度上缓解了慢速的人工操作与快速的计算机运算之间的矛盾。单道批处理系统的特点如下。

- (1) 单道性。一次只能将一个作业装入内存中运行。
- (2) 顺序性。磁带上的一批作业是按其顺序装入内存的。
- (3) 自动性。在监督程序的控制下作业自动执行, 无须人工干预。

单道批处理阶段经历了联机批处理阶段和脱机批处理阶段。

第一, 联机批处理系统。其工作方式如图 1-5 所示。联机批处理提高了计算机的自动化程度, 减少了人工干预。但是快速 CPU 和慢速 I/O 设备之间的串行工作方式, 造成了 CPU 资源的浪费。

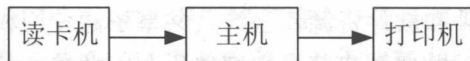


图 1-5 联机批处理系统的工作方式

第二, 脱机批处理系统。其工作方式如图 1-6 所示。脱机批处理系统增设了卫星机, 卫星机同时可与多台外设交互, 主机不再直接与慢速的 I/O 设备进行操作。因此, 脱机批处理方式缓解了 CPU 与 I/O 设备之间的矛盾, 提高了 CPU 的资源利用率。但是 CPU 与外围计算机完全隔离, 可能造成系统“死机”。

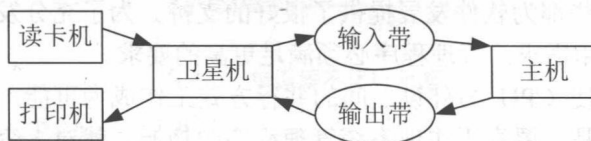


图 1-6 脱机批处理系统的工作方式

批处理操作系统的特点是系统资源利用率高, 作业吞吐量较大, 但是作业的周转时间较长, 且不允许用户任意修改, 即交互性能差, 不利于程序的开发与调试。

2. 分时操作系统

批处理操作系统缺乏交互性, 从而使程序难以在线调试和排错, 因此促成了分时操作系统的产生。分时操作系统(Time Sharing Operating System)的工作方式是: 一台主机连接了若干个终端, 每个终端有一个用户在使用。用户交互式地向系统提出命令请求, 系统接受每个用户的命令, 采用时间片轮转方式处理服务请求, 并通过交互方式在终端上向用户显示结果。用户根据上步结果发出下道命令。分时操作系统将 CPU 的时间划分成若干个

片段,称为时间片(time slice)。操作系统以时间片为单位,轮流为每个终端用户服务。每个用户轮流使用一个时间片,使每个用户感觉像独占计算机系统一样。

分时操作系统的特点如下。

(1) 同时性。同时有多个用户使用一台计算机,宏观上看是多个人同时使用一个CPU,微观上是多个人在不同时刻轮流使用CPU。

(2) 交互性。用户根据系统响应结果进一步提出新请求(用户直接干预每一步)。

(3) 独占性。用户感觉不到计算机为其他人服务,就像整个系统为其所独占。

(4) 及时性。系统对用户提出的请求及时响应。它支持位于不同终端的多个用户同时使用一台计算机,彼此独立,互不干扰,使用户感觉似乎计算机只为自己服务。

常见的通用操作系统是分时系统与批处理系统的结合。其原则是:分时优先,批处理在后。“前台”响应需要频繁交互的作业,如终端的要求,“后台”处理时间性要求不强的作业。

分时操作系统与多道程序系统的差异如下。

(1) 分时操作系统中多个进程轮流使用处理器,每个进程在允许的时间片内可以使用CPU,而在时间片结束时必须释放CPU给系统中的其他进程使用,系统调度程序选中下一个使用CPU的进程,而放弃CPU的进程只能等待下一次使用CPU的机会,直到进程执行完毕。

(2) 因时间片而导致CPU使用权的切换比多道程序系统中要频繁得多,因为分时系统需要保证多个用户在较短的时间内均被响应,以保证用户的使用体验;而多道程序系统中强调的却是效率,单个程序持续占用CPU的时间会长得多。

(3) 所谓多道程序系统,指的是允许多个程序同时进入一个计算机系统的主存储器并启动进行计算的方法。也就是说,计算机内存中可以同时存放多道(两个以上相互独立的)程序,它们都处于开始和结束之间。从宏观上看是并行的,多道程序都处于运行中,并且都没有运行结束;从微观上看是串行的,各道程序轮流使用CPU,交替执行。引入多道程序设计技术的根本目的是提高CPU的利用率,充分发挥计算机系统部件的并行性,现代计算机系统都采用了多道程序设计技术。

(4) 分时操作系统是使一台计算机同时为几个、几十个甚至几百个用户服务的一种操作系统。把计算机与许多终端用户连接起来,分时操作系统将系统处理机时间与内存空间按一定的时间间隔,轮流地切换给各终端用户的程序使用。由于时间间隔很短,每个用户的感受就像独占计算机一样。分时操作系统的特点是可有效增加资源的使用率。

3. 实时操作系统

虽然多道程序系统和分时操作系统能使系统资源利用率得到极大的提升,能带给用户较好的使用体验,但是在一些特殊领域应用时,却无法同时满足实时控制和实时信息处理的需求,于是导致实时操作系统的产生。

实时操作系统(Real-Time Operating System, RTOS)是指使计算机能及时响应外部事件的请求,在规定的严格时间内完成对事件的处理,并控制所有实时设备和实时任务协调一致地工作的操作系统。实时操作系统要追求的目标是:对外部请求在严格时间范围内做出反应,有高可靠性和完整性。其主要特点是资源的分配和调度首先要考虑实时性,然后才