

卓越工程师教育培养计算机类创新系列规划教材

操作系统 设计原理

主 编 鞠世光 詹永照
副主编 薛安荣 蔡 涛 牛德姣 毛启荣



科学出版社

卓越工程师教育培养计算机类创新系列规划教材

操作系统设计原理

主 编 鞠时光 詹永照

副主编 薛安荣 蔡 涛 牛德姣 毛启容

科学出版社

北 京

内 容 简 介

本书系统地介绍操作系统设计的基本概念、原理和实现方法, 主要内容包括: 操作系统概述、进程与线程、互斥与同步、处理器调度、存储器管理、文件管理、输入/输出管理、操作系统安全、新型操作系统共 9 章。在每章的基本原理和方法介绍后, 均以目前最流行的操作系统 Linux、UNIX 和 Windows 为例, 介绍当代操作系统的实现原理和特点。最后一章专门介绍嵌入式操作系统、分布式操作系统和虚拟技术等目前新型的操作系统。

本书可作为计算机类本科专业的教材, 也可供有关专业的教师、学生和科技人员阅读参考。

图书在版编目(CIP)数据

操作系统设计原理/鞠时光, 詹永照主编. —北京: 科学出版社, 2016.1
卓越工程师教育培养计算机类创新系列规划教材
ISBN 978-7-03-047100-0

I. ①操… II. ①鞠… ②詹… III. ①操作系统-程序设计-教材
IV. ①TP316

中国版本图书馆 CIP 数据核字(2016)第 012048 号

责任编辑: 邹 杰 张丽花/责任校对: 桂伟利

责任印制: 霍 兵/责任设计: 迷底书装

科学出版社出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

保定市中美美凯印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2016年1月第 一 版 开本: 787×1092 1/16

2016年1月第一次印刷 印张: 20 1/2

字数: 486 000

定价: 48.00 元

(如有印装质量问题, 我社负责调换)

前 言

操作系统是计算机系统最基本的系统软件之一，是用户开发和使用应用软件不可缺少的支撑环境。随着计算机系统软硬件规模的日益扩大和性能的不断提高，用户与操作系统的联系愈加密切，操作系统课程是计算机类专业的必修课程之一。

本书改变了传统教材的框架，把操作系统的基本原理与实践应用有机地结合起来，以原理指导应用，又从应用中加深对原理的理解。在介绍操作系统基本设计原理的同时，与 UNIX、Linux、Windows 等常用操作系统的实现技术和特点相结合，做到有利于学生对这些常用操作系统的理解和实际应用能力的培养。

全书共 9 章，其内容覆盖学生学习操作系统课程应掌握的基本概念、基本原理、主要技术和基本操作。其中，第 1 章主要介绍操作系统的基本概念、发展历史及操作系统的分类。第 2 章主要介绍进程、线程及多核多线程的概念。第 3 章主要介绍互斥与同步的基本概念、进程互斥和同步的算法设计，以及进程通信的实现方式和死锁及其解决方法。第 4 章主要介绍处理器调度类型、单/多处理器调度算法和实时调度算法。第 5 章主要介绍几种常用的存储管理方法，如分区存储管理、页式存储管理、段式存储管理、段页式存储管理和虚拟存储器等原理和实现方法。第 6 章主要介绍文件系统概念、文件组织、文件目录、文件存储、文件共享与文件系统的一致性。第 7 章主要介绍 I/O 系统的结构、控制、缓冲、软件设计及磁盘调度算法。第 8 章首先介绍计算机安全的一些基本概念，接着从安全体系结构的设计和实现方法等方面来阐述操作系统的安全性，并给出研究和开发安全操作系统的实例。第 9 章主要介绍一些新型的操作系统，包括嵌入式操作系统和分布式操作系统两大类，并介绍云计算和大数据中非常重要的虚拟化技术。本书各章均穿插介绍流行的操作系统相关设计与实现方法，同时章前均有知识要点、预习准备、兴趣实践和探索思考，章后有小结、拓展阅读、复习思考题及上机实验，便于学生探索学习和教师按需授课。

参加编写的人员及分工：第 1 章由鞠时光编写，第 2、3 章由牛德姣编写，第 4 章由毛启容编写，第 5 章由詹永照编写，第 6、7 章由薛安荣编写，第 8、9 章由蔡涛编写。全书由鞠时光、詹永照统一规划与统稿。

由于书中所涉及的操作系统 UNIX、Linux 和 Windows 只是作为应用实例来介绍的，所以这些部分难免不成体系。需要系统学习这些操作系统的实现及使用方法的读者，可以进一步参考有关资料。另外，操作系统的发展日新月异，加之编者水平有限，书中不当之处在所难免，敬请读者批评指正。

编 者

2015 年 9 月

目 录

前言

第 1 章 操作系统概述.....1

1.1 操作系统的定义.....1

1.1.1 基本概念.....1

1.1.2 一个计算机系统的视图.....2

1.1.3 操作系统的基本功能.....3

1.2 操作系统的形成和发展.....4

1.3 操作系统的分类.....6

1.3.1 单用户操作系统.....6

1.3.2 批处理操作系统.....7

1.3.3 实时操作系统.....7

1.3.4 分时操作系统.....8

1.3.5 网络操作系统.....8

1.3.6 分布式操作系统.....8

1.3.7 嵌入式操作系统.....9

1.4 操作系统的运行环境.....9

1.4.1 中央处理器(CPU).....10

1.4.2 特权指令.....10

1.4.3 处理器的状态.....10

1.4.4 程序状态字(PSW).....10

1.5 操作系统的结构.....11

1.6 现代操作系统.....12

1.6.1 现代操作系统的特点.....12

1.6.2 UNIX 的特点.....14

1.6.3 Linux 的特点.....15

1.6.4 Windows NT 的特点.....16

本章小结.....19

习题.....20

实验.....20

第 2 章 进程与线程.....21

2.1 多道程序与并发执行.....21

2.1.1 单道程序的顺序执行.....21

2.1.2 多道程序的并发执行.....22

2.2 进程模型.....24

2.2.1 进程的概念.....24

2.2.2 进程的实体.....26

2.2.3 进程状态和转换.....29

2.2.4 进程控制.....30

2.3 线程模型.....32

2.3.1 线程的概念.....32

2.3.2 线程与进程的比较.....34

2.3.3 线程的实现.....36

2.4 多核、多线程与超线程.....37

2.4.1 多核技术简介.....37

2.4.2 多核基本概念.....39

2.5 进程管理实例.....42

2.5.1 UNIX 进程管理.....42

2.5.2 Linux 进程管理.....46

2.5.3 Windows 进程管理.....49

本章小结.....53

习题.....53

实验.....54

第 3 章 互斥与同步.....55

3.1 进程管理的背景.....55

3.2 进程互斥.....58

3.2.1 互斥与临界区.....58

3.2.2 互斥的软、硬件实现方法.....59

3.2.3 信号量及 P、V 原语.....63

3.2.4 用 P、V 操作实现互斥.....65

3.3 进程同步.....66

3.3.1 进程同步概念.....66

3.3.2 用 P、V 操作实现同步.....67

3.4 进程通信.....69

3.4.1 进程通信的类型.....69

3.4.2 消息传递.....70

3.4.3 进程通信的有关问题.....73

3.5 死锁.....74

3.5.1 死锁的概念.....74

3.5.2 死锁的必要条件	75	4.6.2 对称多处理器上的线程调度	118
3.5.3 死锁的防止	75	4.7 UNIX 多处理器调度与实时调度方法	119
3.5.4 死锁的避免	77	4.8 Linux 处理器调度方法	120
3.5.5 死锁检测与恢复	78	4.8.1 实时调度	120
3.6 经典问题	79	4.8.2 非实时调度	122
3.6.1 读者-写者问题	79	本章小结	122
3.6.2 哲学家进餐问题	81	习题	123
3.6.3 打瞌睡的理发师问题	82	第5章 存储器管理	125
3.7 多核环境下的进程同步	84	5.1 存储管理的功能	125
3.7.1 多核进程同步	84	5.1.1 计算机系统的多级存储结构	125
3.7.2 硬件原子操作	85	5.1.2 存储管理的任务和功能	126
3.7.3 总线锁	85	5.2 存储分配的几种形式与重定位	127
3.7.4 多核环境下的软件同步原语	85	5.2.1 存储分配的几种形式	127
3.7.5 旋锁	86	5.2.2 重定位	128
3.8 进程同步与通信实例	88	5.2.3 覆盖与交换	130
3.8.1 UNIX 同步与通信	88	5.3 单道环境下的存储管理	131
3.8.2 Linux 进程通信	91	5.4 分区存储管理	132
3.8.3 Windows 同步与通信	92	5.4.1 固定分区法	132
本章小结	95	5.4.2 动态分区法	133
习题	96	5.4.3 地址转换与存储保护	136
实验	98	5.4.4 分区的共享	137
第4章 处理器调度	99	5.4.5 移动技术	137
4.1 处理器调度层次	99	5.4.6 分区存储管理的优缺点	138
4.1.1 长程调度(Long-Term Scheduling)	100	5.5 页式存储管理	138
4.1.2 中程调度(Medium-Term Scheduling)	100	5.5.1 概述	138
4.1.3 短程调度(Short-Term Scheduling)	101	5.5.2 静态页式存储管理	139
4.2 调度准则	101	5.5.3 虚拟页式存储管理	144
4.3 短程调度算法	102	5.5.4 页式存储管理的优缺点	156
4.3.1 单处理器短程调度算法	102	5.6 段式及段页式存储管理	156
4.3.2 多处理器短程调度算法	107	5.6.1 段式存储管理	156
4.4 实时调度	112	5.6.2 段式虚拟存储管理	158
4.4.1 实时调度算法的特点	112	5.6.3 段页式虚拟存储管理	161
4.4.2 实时调度算法的分类	113	5.7 内存管理实例	164
4.4.3 常用的实时调度算法	114	5.7.1 UNIX 内存管理	164
4.5 传统 UNIX 单处理器调度方法	115	5.7.2 Linux 内存管理	166
4.6 Windows 2000/XP 处理器调度方法	117	5.7.3 Windows 内存管理	169
4.6.1 线程优先级	117	本章小结	171
		习题	172

实验	173
第 6 章 文件管理	174
6.1 文件	174
6.1.1 文件命名	174
6.1.2 文件类型	175
6.1.3 文件属性	175
6.1.4 文件存取方法	177
6.1.5 文件操作	177
6.1.6 文件的逻辑结构	177
6.2 文件目录	181
6.2.1 文件目录内容	181
6.2.2 文件目录结构	182
6.3 文件共享与保护	187
6.3.1 文件共享	187
6.3.2 文件保护	189
6.3.3 文件系统的一致性	190
6.4 文件存储空间的管理	191
6.4.1 空闲表法	191
6.4.2 空闲链表法	192
6.4.3 位示图法	192
6.4.4 链接索引表法	193
6.5 文件分配	193
6.5.1 连续分配	194
6.5.2 链接分配	195
6.5.3 索引链接分配	196
6.6 文件管理实例: UNIX 文件管理	197
6.6.1 UNIX 文件类型	198
6.6.2 UNIX 文件系统存储结构	198
6.6.3 目录与索引结点	199
6.6.4 文件分配	200
6.6.5 文件系统的打开与读写	202
6.7 文件管理实例: Linux 虚拟文件 系统	203
6.7.1 超级块对象	206
6.7.2 索引结点对象	207
6.7.3 目录项对象	208
6.7.4 文件对象	209
6.7.5 主要的数据结构之间的关系	210
6.7.6 Linux 文件系统的安装和管理	210

6.8 文件管理实例: Windows 文件管 理 (NTFS)	211
6.8.1 NTFS 的主要特性	211
6.8.2 NTFS 卷和文件结构	212
6.8.3 NTFS 的可恢复性	214
本章小结	215
习题	216
实验	217
第 7 章 I/O 管理	218
7.1 I/O 管理概述	218
7.1.1 I/O 管理目标与功能	218
7.1.2 I/O 系统组成	219
7.2 I/O 控制方式	224
7.2.1 程序直接控制方式	225
7.2.2 中断控制方式	225
7.2.3 DMA 控制方式	226
7.2.4 通道控制方式	227
7.3 I/O 缓冲	230
7.3.1 单缓冲	231
7.3.2 双缓冲	231
7.3.3 循环缓冲	232
7.3.4 缓冲池	232
7.3.5 缓冲的作用	234
7.4 设备驱动程序	234
7.4.1 设备驱动程序的模式	234
7.4.2 设备驱动程序的功能	234
7.4.3 设备驱动程序的特点	235
7.4.4 设备驱动程序的处理过程	235
7.5 设备分配	236
7.5.1 设备分配原则与分配方式	236
7.5.2 设备分配时应考虑的因素	236
7.5.3 设备分配中的数据结构	238
7.5.4 I/O 设备分配的基本流程	239
7.5.5 SPOOLing 技术	240
7.6 磁盘存储器的管理	241
7.6.1 磁盘性能参数	243
7.6.2 磁盘调度	243
7.7 磁盘阵列 (RAID)	246
7.7.1 原理	246

7.7.2 优点	247	8.3.7 备份与容错	284
7.7.3 标准 RAID	247	8.3.8 隐通道分析与处理	284
7.8 I/O 管理实例: UNIX I/O 管理	251	8.4 安全模型	285
7.8.1 缓冲区管理	251	8.4.1 Bell-LaPadula 模型	285
7.8.2 设备驱动程序的接口	253	8.4.2 Biba 模型	286
7.9 I/O 管理实例: Linux I/O 管理	254	8.4.3 Clark-Wilson 模型	286
7.9.1 Linux 网络设备	254	8.4.4 中国墙模型	287
7.9.2 页面缓存 (page cache)	254	8.4.5 RBAC 模型	287
7.9.3 页面缓存的预读	256	8.5 操作系统安全体系结构	288
7.9.4 Linux 硬盘管理	258	8.5.1 安全体系结构的含义	288
7.9.5 Linux 设备驱动程序	259	8.5.2 FAM 框架	289
7.10 I/O 管理实例: Windows I/O 管理	259	8.5.3 Flask 体系	289
7.10.1 I/O 的系统结构	260	8.6 安全操作系统简介	290
7.10.2 I/O 管理器	260	8.6.1 SE-Linux	290
7.10.3 I/O 函数	261	8.6.2 EROS	291
7.10.4 设备驱动程序	262	8.6.3 红旗安全 Linux	293
7.10.5 I/O 处理	262	本章小结	298
本章小结	262	习题	298
习题	263	实验	298
实验	264	第 9 章 新型操作系统	299
第 8 章 操作系统安全	265	9.1 嵌入式操作系统	299
8.1 操作系统安全概述	265	9.1.1 EOS 简介	300
8.1.1 计算机系统安全	265	9.1.2 微软嵌入式操作系统	300
8.1.2 操作系统安全	267	9.1.3 嵌入式 Linux 与 Android	302
8.1.3 基本概念	268	9.1.4 TinyOS	303
8.2 安全评估与标准	269	9.1.5 VxWorks	304
8.2.1 美国国防部的“橙皮书”(TCSEC)	270	9.1.6 嵌入式操作系统的发展方向	305
8.2.2 欧洲安全评价标准 (ITSEC)	271	9.2 分布式操作系统	307
8.2.3 国际通用安全评价标准 (CC)	272	9.2.1 多核操作系统	307
8.2.4 加拿大安全评测标准 (CTCPEC)	272	9.2.2 客户机/服务器计算模型	309
8.2.5 中国推荐标准 GB/T 18336—2001	273	9.2.3 集群系统	310
8.3 安全机制	273	9.2.4 MapReduce 计算模型	312
8.3.1 标识与鉴别机制	273	9.3 虚拟化技术	315
8.3.2 访问控制	276	本章小结	319
8.3.3 监控与审计机制	278	习题	319
8.3.4 存储保护、运行保护和 I/O 保护	279	实验	319
8.3.5 加密	280	参考文献	320
8.3.6 恶意代码	281		

第1章 操作系统概述

知识要点：主要包括操作系统的定义、操作系统的形成和发展、操作系统的分类、操作系统的运行环境、操作系统的结构和现代操作系统的特征。要求粗略了解支撑操作系统执行的核心硬件以及操作系统的层次结构。初步理解不同类型操作系统之间的异同。熟悉操作系统的发展概况。

预习准备：回想平时接触和使用的不同类型操作系统，思考这些操作系统之间的共同特征与区别。

兴趣实践：收集整理所接触和了解的操作系统，在后续学习中比较这些不同类型操作系统各方面的异同。

探索思考：思考书中给出的操作系统定义、分类和结构与自己日常认识的差异，思考不同应用领域对操作系统的不同需求。

操作系统是计算机系统的一种大型系统软件。本章将给出操作系统的定义，分析操作系统的功能与类型，介绍操作系统的形成和发展，并对操作系统运行的硬件环境有选择地进行介绍。本章的最后介绍了 UNIX、Linux 以及 Window NT 操作系统的特点。

1.1 操作系统的定义

一个完整的计算机系统，不论是大型机、小型机，还是微型机，都是由硬件和软件两大部分组成。通常硬件是指计算机物理装置本身，也就是指计算机的各种处理器(如中央处理器)、存储器、I/O 设备和通信装置等。软件是相对于硬件而言的，它是指由计算机硬件执行以完成一定任务的程序、数据及有关文档。

1.1.1 基本概念

计算机软件通常分成两大类，即系统软件和应用软件。系统软件用于计算机的管理、维护、控制和运行以及对运行的程序进行翻译、装入等服务工作。系统软件本身又可分成三部分，即操作系统、语言处理系统和常用的例行服务程序。语言处理系统包括各种语言的编译程序、解释程序和汇编程序。服务程序的种类很多，通常包括库管理程序、连接编辑程序、诊断排错程序等。应用软件是指那些为了某一类应用需要而设计的程序、或用户为解决某个特定问题而编制的程序或程序系统。

操作系统是计算机系统的一种系统软件，它用于管理计算机的资源和控制程序的执行。一个程序只有在通过操作系统获得必需的资源后才能执行。例如，程序在执行前必须获得主存储器资源才能装入，它的执行要依靠处理器，它在执行中还可能要用外部设备输入或输出数据，或者使用计算机系统中的文件以及调用子程序等。计算机配置了操作系统后可

以提高效率，便于使用。现在，操作系统已成为计算机系统中不可缺少的一种系统软件。

1.1.2 一个计算机系统的视图

一个计算机系统可以被认为是由硬件和软件按层次方式构成的。图 1-1 表示了一个四层结构，每层表示一组功能和一个接口。接口是用于在该层内实现功能的一组可见的约定，我们把接口的这些特性称为计算机系统的一个视图。

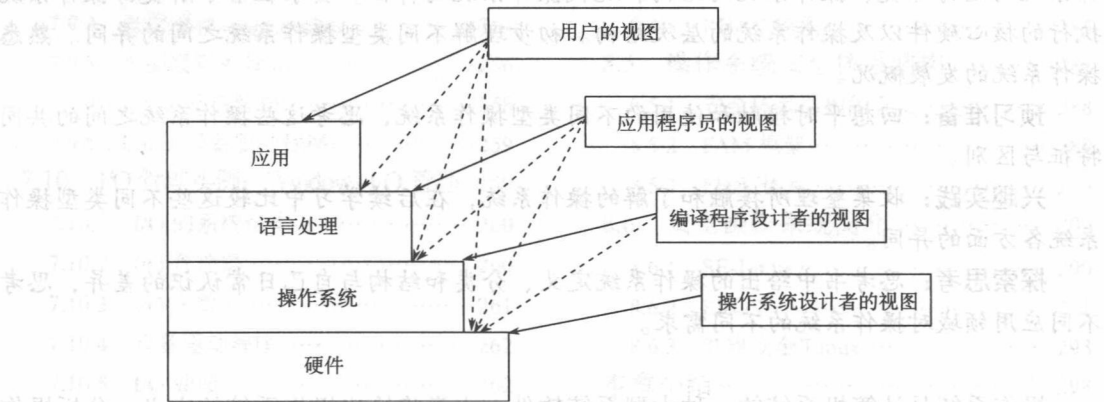


图 1-1 计算机系统的分层和视图

由图 1-1 可见，硬件是软件建立与活动的基础，而软件是对硬件功能的扩展。操作系统是核心的系统软件，与硬件的关系尤为密切，它不仅对硬件资源直接实施控制、管理，而且很多功能的完成是与硬件动作配合实现的。所以操作系统的运行必须依靠良好的硬件支撑环境。

硬件层表示机器的可见结构，它包括可执行一组指令的处理器，若干个供程序使用的寄存器和用于访问存储器的寻址模式，还包含诸如通道、控制器、处理器和存储器之间的关系。它是操作系统工作的基础，因此，对操作系统的设计者来说，他所看见的系统视图就是硬件层。

操作系统对硬件层作第一次功能扩充，以便为编译程序的设计者和应用程序员提供有效的服务。它提供接口以便容易地开发系统程序。操作系统是整个计算机系统的控制管理中心，其中也包括对其他各种软件的控制和管理，如编辑程序、编译程序、连接装配程序、数据库系统和各种软件工具等。操作系统对它们既具有支配权力，又为其运行建造环境。操作系统提供的接口并不能完全隐藏硬件特性，因此，一个编译程序的设计者可能需要某些机器特性的知识。编译工作的基础是被操作系统扩充了功能的机器，它由软件定义的操作系统接口和硬件指令集合的某些部分组成。类似地，一个使用汇编语言的程序员将利用操作系统和硬件提供的复合功能。因此，向编译程序的设计者所展示的系统视图除了操作系统外，还应加上操作系统未能隐藏的硬件特性，它们在图 1-1 中分别用实线箭头与虚线箭头表示。

同样，应用程序员的视图除了语言处理层外，还有未被隐藏的部分操作系统和硬件的

特性。而用户的视图除了应用层外，还有未被隐藏的部分语言处理层、操作系统层及硬件层的特性。它们在图 1-1 中也分别用实线箭头和虚线箭头表示。

用户包括几种不同身份的人：一般用户、操作员和管理员等。一般用户是计算机系统环境中的顾客，他们利用计算机来完成各种有用的任务。操作系统对外提供的功能就是它与一般用户之间的接口。操作员是负责启动系统、监督系统状态、实施对设备(如磁带、打印机)操作的人员，他们是对系统的最终控制者。管理员负责制定系统调度政策，确立维修、改进方案。

1.1.3 操作系统的基本功能

前面说明了操作系统的基础地位。其实，大家也会体验到，使用计算机需首先装入操作系统。计算机配置了操作系统后可以提高效率，便于使用。可以说，它是计算机系统的一个聪明能干的“大管家”。下面从两方面简要介绍操作系统所需提供的基本功能。

1. 人机交互界面

通常操作系统需要提供人机界面功能为用户提供各种计算服务。用户可直接使用键盘命令或 Shell 命令语言或图形用户界面(GUI)与计算机交互，表达计算要求，由操作系统人机交互界面功能模块分析处理并调用系统内部功能模块(系统调用)，最终完成用户的计算任务并将结果反馈。这两项是操作系统与用户联系的软接口，通常称为“界面”。它和实用层软件一起，提供了用户开发和运行应用软件的环境和手段。这是操作系统为用户提供的基本公共服务。

事实上，用户若没有操作系统的支持服务，即使是简单的操作，也显得十分吃力。例如用户想显示存在磁盘上的一个文件 FILE.TXT。借助 UNIX 操作系统完成这项操作，只须键入命令：`$cat FILE.TXT` 即可实现，十分简便。然而，若无操作系统支持，则需要用户使用难记难认的机器代码编写 4 个模块以上的程序，如输入程序、读盘和搜索文件程序、装入内存程序和驱动显示程序。这些程序中涉及内存、外设中若干物理特性和状态参数，还需要字符库和调试软件的支持。对于一般用户，做到这些是难以想象的。有了操作系统的服务，用户便可驾轻就熟了。如今，大部分操作系统都提供了图形用户界面功能，用户操作计算机是一种所见即所得的计算实现方式，计算机的操作就更加方便了。应该说，目前微机系统推广应用得如此之广，与它配置了功能强、使用简便的操作系统人机交互界面有着密切的关系。

2. 资源管理

使用计算机，除方便用户以外，尚需使它高效率地运转。为此，需要合理地分配使用各种软、硬件资源。如使慢速的外设与高速的 CPU 相匹配的问题；内存空间紧张、如何发挥外部存储器效率的问题；众多的文件信息如何组织、存取的问题。

从资源管理的观点来看，操作系统的功能主要包括：处理器与进程管理、存储器管理、文件管理、设备管理和网络管理。

(1) 处理器与进程管理。处理器与进程管理功能包括：调度、进程控制、进程同步与进程通信等。

调度包括作业调度和进程调度。作业调度是考虑充分利用系统资源的要求,将用户的算题按照一定的策略,为它分配资源,调入内存并创建进程,使之有机会获得系统的服务。通常在大中型操作系统中,才提供作业管理的功能。一般微机操作系统以及单用户操作系统不考虑作业管理的功能。进程调度就是考虑何时为进程分配处理器并占有多长时间,让它真正占有处理器进行算题。进程控制就是控制进程的运行状态的变化,让进程有序地走走时停。进程同步就是解决进程之间使用资源的竞争和协作问题。进程通信就是使进程之间由于算题需要能够传输相关信息。

(2) 存储器管理。存储器管理主要管理主存储器资源。存储管理将根据用户程序的要求给它分配主存储器,并将程序的逻辑地址空间转换为物理地址空间,同时还要考虑内存空间的共享和保护,用户存放在主存储器中的程序和数据不被破坏,此外还要考虑如何使有限的内存能够运行更多和更大的程序,即主存扩充问题。操作系统的这一部分功能与硬件存储器的组织结构密切相关,操作系统的设计者应根据硬件情况和使用需要,采用各种相应的有效调度策略与保护措施。

(3) 文件管理。文件管理支持对文件的存储、检索和修改等操作以及文件保护的功能。早期的管理程序仅提供一个简单的文件系统,而现代的操作系统一般都提供功能复杂的文件系统,多数还提供数据库系统来实现信息的管理工作。

(4) 设备管理。设备管理负责管理各类外围设备,包括分配、启动和故障处理等。为了提高效率,还引入了逻辑(虚拟)设备的概念,以实现预输入和缓输出功能。

(5) 网络管理。目前的多数系统都有联网计算要求,因此操作系统还要提供网络资源管理功能,以实现信息的网络传输、网络资源服务和网络安全防护等目的。

1.2 操作系统的形成和发展

第一代计算机运行速度较低,外围设备较少,因而,编制和运行一个程序也比较简单。那时,程序员往往直接使用机器语言来编制一个程序,这种“目标程序”被人为地穿在卡片(或纸带)上,并用一个引导程序装入主存储器。程序员通过控制台开关来调试和操作运行程序。在这期间,整个计算机都被一个程序员所占有。因而,不需要专门的操作员,程序员身兼两职——既是操作员,也是程序员。

随着计算机的发展,协助用户使用计算机软件——原始汇编系统产生了。在这样的系统中,数字操作码被记忆码所代替,程序按一个固定格式的汇编语言书写。程序员(或系统程序员)预先编制一个汇编解释程序,它把汇编语言书写的“源程序”解释成计算机能直接执行的机器语言表达的“目标程序”。因而,在这样的计算机系统中,首先需要把这个汇编解释程序和源程序都穿在卡片或纸带上,然后再装入和执行。如图 1-2 所示,整个计算分两个阶段,六个计算步,每个计算步的功能如下:

(1) 通过引导程序把汇编解释程序装入到计算机中。

(2) 通过汇编解释程序读入源程序,并执行汇编过程。

(3) 产生一个目标程序,并输出到卡片或纸带上。

(4) 通过引导程序把目标程序装入计算机。

用来 (5) 目标程序读入卡片数据或纸带数据。 (6) 产生计算结果, 并输出到卡片或打印纸上。
其中 (1)~(3) 的三个计算步是汇编阶段, (4)~(6) 的三个计算步是执行阶段。



图 1-2 原始汇编和执行过程

到了 20 世纪 60 年代, 硬件技术取得了两个方面的重大进展: 一是通道技术的引进, 二是中断技术的发展, 再加之存储容量的增长, 这就给软件的发展奠定了物质基础。在这期间, 先后出现了 FORTRAN 和 ALGOL 等程序设计语言与相应的编译程序以及程序库的使用等。同时, 出现了对计算机硬件和软件进行管理与调度的软件, 管理程序, 即初级的操作系统。

管理程序的主要功能是: 向用户提供多个共享资源来运行他们的程序; 帮助操作员控制用户程序的执行和管理计算机的部分资源。

有了管理程序以后, 用户不必亲自上机操作, 而是由专业化的操作员代劳。操作员只要通过控制台打字机打入控制命令就可以操纵计算机, 操作员打入的命令由管理程序来识别和执行。这样, 不仅操作速度快, 而且操作员可以方便地进行一些较为复杂的控制。当计算机运行中发生错误或意外时, 管理程序通过计算机从控制台打字机上输出信息向操作员报告。这种输出信息不仅比“亮灯显示”所表达的更为丰富, 而且操作员也易于理解。总之, 用这种半自动方式来控制计算机不仅提高了效率, 而且方便了使用。

到了第二代计算机后期, 特别是进入第三代以后, 软件有了很大的发展, 它的作用也日益显著。同时, 硬件也有了很大的发展, 特别是主存储器容量的增加和大容量辅助存储器“磁盘”的出现, 给发展更先进的管理程序准备好了物质条件。另外, 计算机应用的日益广泛和深入, 也要求进一步发展和扩大功能简单的管理程序。这样, 管理程序就迅速地发展成为一个主要的软件分支: 操作系统。

随着大规模集成电路技术的发展, 微型计算机迅速地发展起来。从 20 世纪 70 年代中期开始, 出现了微型机操作系统。1970 年, 美国 Digital Research 软件公司研制了操作系统

CP/M, 由于它短小、精致和适应性强, 因而, 此后出现的一些 8 位微型机操作系统多采用 CP/M 的结构。

随着微型计算机及以微型计算机为其主要节点的局域网的发展, 操作系统的研究、开发、生产与销售也获得了飞速的发展。微型机操作系统的发展大致经历了两个阶段: 第一阶段(1976~1979 年)为单用户、单道作业的操作系统。继 CP/M 之后, 还有 CDOS(Cromemco 磁盘操作系统)、MDOS(Motorola 磁盘操作系统)、TRSDOS(TRS 磁盘操作系统)、SDOS(SD 磁盘操作系统)和 MS-DOS(Microsoft 磁盘操作系统)。第二阶段(1979~1980 年)为多用户、多道作业和分时系统, 例如: MP/M(多用户监控程序)、AMOS 和 XENIX。20 世纪 90 年代以来, Windows、Linux、UNIX 操作系统几乎垄断了微机操作系统的市场。

微型计算机和 Internet 的普及应用催生了软件巨头 Microsoft。当今, 物联网与云计算的兴起, 无论何时、何地、何物, 只要需要就能实现针对性计算, 这给计算系统的核心软件操作系统的研究与开发带来了更大的挑战, 新一代的操作系统应运而生。为了方便便携式终端、传感器和物件接入点开发各种应用系统, 产生了嵌入式操作系统。

云计算是一种通过 Internet 以服务方式提供动态可伸缩的虚拟化资源的计算模式。为了实现云计算模式, 必须研发云计算操作系统。云计算操作系统是指构架于服务器、存储、网络等基础硬件资源和单机操作系统、中间件、数据库等用于管理海量的基础硬件、软件之上的云平台综合管理系统。它是一个大型复杂的操作系统。由于云计算的需要, 基于网络的分布式操作系统和虚拟化软件的研究开发又唤来了新的春天。可以想象, 将来微型化和大型化这两大类操作系统必将成为市场的宠儿。

1.3 操作系统的分类

当前, 计算机应用已广泛地深入到人类生活的各个领域。在这些应用中, 人们对计算机的要求不尽相同, 对计算机操作系统的性能要求、使用方式也是十分不同的, 因此, 对操作系统的类型进行分类的方法也很多。比如, 可按硬件系统的大小、系统的属性和用户的属性等来进行分类。在此, 我们按照操作系统所提供的功能来进行分类, 把操作系统大致可以分成以下 7 类: 单用户操作系统、批处理操作系统、实时操作系统、分时操作系统、网络操作系统、分布式操作系统、嵌入式操作系统。下面分别介绍这几类操作系统的特性。

1.3.1 单用户操作系统

简单地说, 一个操作人员在一个终端上使用计算机就是一个用户。早期的个人微机, 由一个主机带一个终端, 同一时间只能为一个用户服务, 使用的是单用户操作系统。单用户操作系统的根本特征是, 一个用户独占计算机系统资源。系统所有软、硬件资源全为一个用户服务, 单独地执行该用户提交的一个任务。

例如, IBM-PC 个人微机和兼容机配的 MS-DOS, 以及 8 位机使用的 CP/M 操作系统均属于单用户操作系统。使用过程中, 即便多数资源是空闲的, 也被一个用户所独占。显然系统资源未能充分利用。但其操作系统简单, 易被人们掌握。

1.3.2 批处理操作系统

在一般计算中心(或数据中心)的小型以上的计算机上所配置的操作系统通常属于批处理操作系统。用户把要计算的问题、数据和作业说明书一起交给操作员,操作员将一批算题输入到计算机,然后由操作系统来控制执行。通常,采用这种批量化处理作业技术的操作系统称为批处理操作系统。

批处理操作系统又分为单道和多道批处理系统,它们的区别在于:

(1)作业道数。单道批处理系统中只有一道作业在主存中运行。而多道批处理系统中同时有多道作业在运行。

(2)作业处理方式。单道批处理系统是把多个用户作业形成一批,由卫星机将这些作业输入磁带中,然后主机再从该磁带中将作业一个一个地读入主存进行处理。作业完成后,将结果也都输出到另一磁带中去,当这批作业全部完成后,再由卫星机把此磁带上的结果通过相应的输出设备输出。处理完一批作业后再处理另一批作业。而在多道批处理系统中(包括网络中的远程批处理),作业可随时(不必集中成批)被接受进入系统,并存放在磁盘输入池中形成作业队列。而后操作系统按一定原则从作业队列中调入一个或多个作业进入主存运行。所以,“批”的概念已不十分明显。这里所谓的“批处理”是指这样一种操作方式:用户与他的作业之间没有交互作用,不能直接控制其作业的运行。一般称这种方式为批操作。

IBM DOS(磁盘操作系统)是一个典型的批处理多道系统。它是一个通用操作系统,开始是为 IBM / 360 的较小型号设计的,后来扩展到 370 系统。以后它又发展成 DOS / VS 和 DOS / VSE 运行于 IBM43 系列机上。

1.3.3 实时操作系统

“实时”是指对随机发生的外部事件作出及时的响应并对其进行处理。所谓外部是指来自与计算机系统相连的设备所提出的服务和采集数据要求。对一个特殊事件的处理活动是由一串处理任务来完成的,其中每个处理任务必须在规定的时间内完成。外部事件指接收数据或请求一个联机设备服务,其中联机设备并非由操作员来驱动,而是由系统根据外部事件的请求和联机设备当时的状态来确定该联机设备是否对该请求作出响应。

实时系统是较少有人为干预监督和控制的系统。仅当系统内的计算机识别到了违反系统规定的限制或者计算机本身发生故障时,系统才需要人为干预。人为干预允许重置参数和调整监督设备的任务。用于实时控制的计算机系统要确保在任何时候,甚至在满载时都能及时响应。因此,设计实时操作系统时,首先要考虑响应及时,其次才考虑资源的利用率。

实时系统的软件依赖于应用的性质和实际使用的计算机类型。然而,对于实时操作系统而言,它的一个基本特征是事件驱动设计,即当接收了某种类型的外部消息后,由系统选择一个程序去执行。

实时系统的应用十分广泛,例如:监督产品线、流水线生产的连续过程,监督病人的脑界功能,监督和控制交通灯系统,监督和控制实验室的实验以及监督军用飞机的状态等。

1.3.4 分时操作系统

所谓分时操作系统是指多个用户分享使用同一台计算机,也就是说,把计算机的系统资源(尤其是 CPU 资源)进行时间上的分割,即将 CPU 整个工作时间分成一个个的时间段,每个时间段称为一个时间片。让每个用户轮流使用这些时间片。使得每个用户均感到自己独享该 CPU。

分时操作系统的主要目的是对联机用户的服务和响应。它的主要特点为:

(1)同时性。若干个终端用户可同时使用计算机。

(2)独立性。用户彼此独立,互不干扰。

(3)及时性。用户的请求能在较短时间内得到响应。

(4)交互性。用户能进行人机对话,联机地调试程序,以交互方式工作。

分时操作系统和批处理多道操作系统的第一个差别是它们在目标上存在着基本的不同。一个批处理多道程序系统的目标是提高机器效率;而分时操作系统的目标是对用户请求的快速响应。

分时操作系统和批处理多道操作系统的第二个差别表现在提交给系统的作业性质上。对于要求在几分钟内能从终端上获得结果的短小作业来说,分时系统是最有效的;但是,对于需要较长时间才能完成的大型作业而言,批处理多道操作系统较为有效。

分时操作系统和批处理多道操作系统的第三个差别在于:对于充分使用系统资源而言,批处理多道操作系统是较好的,因为它可以同时接收经过合理安排的各种不同负载的作业;对于要求执行相同功能的作业而言,分时系统是较好的,因为在不同的终端上同时使用同一个功能的例行子程序将减少系统调用它的开销。

1.3.5 网络操作系统

计算机网络是通过通信机构把地理上分散且独立的计算机连接起来的一种网络。有了计算机网络之后,用户可以突破地理条件的限制,方便地使用远程的计算机资源,实现资源共享。提供网络通信和网络资源共享功能的操作系统称为网络操作系统。

网络操作系统除了应具有的处理与进程管理、存储器管理、设备管理和文件管理外,还应具有以下两大功能:

(1)提供高效、可靠的网络通信能力。

(2)提供多种网络服务功能,如:远程作业录入并进行处理的服务功能、文件传输服务功能、电子邮件服务功能、远程打印服务功能等。总之,要为用户提供访问计算机网络中各种资源的服务。

1.3.6 分布式操作系统

分布式操作系统是一种特殊的网络操作系统,它是一种用于管理分布式系统资源的操作系统。它与集中式操作系统的主要区别在于资源管理、进程通信和系统结构方面。它是由多台计算机组成的网络并且满足以下条件的系统:

(1)系统中任意两台计算机可以通过通信来交换信息。

(2) 系统中各台计算机无主次之分, 既没有控制整个系统的主机, 也没有受控于他机的从机。

(3) 系统的资源为所有用户共享。

(4) 系统中若干台计算机可以互相协作来完成一个共同任务, 或者说, 一个程序可以分布于几台计算机上并行地运行。

1.3.7 嵌入式操作系统

嵌入式操作系统是一种支持嵌入式应用的操作系统, 它是嵌入式系统极为重要的组成部分。嵌入式操作系统通常包括以下几个模块: 与硬件相关的底层驱动程序、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等。

嵌入式操作系统具备通用操作系统的如下基本特点: ①能够有效管理复杂的系统资源; ②能够对硬件进行抽象; ③能够提供库函数、驱动程序、开发工具集等。

与通用操作系统相比较, 嵌入式操作系统具有的特性如下: ①系统具有实时性; ②系统具有硬件依赖性; ③系统软件具有固化性; ④系统具有应用专用性; ⑤系统具有可裁剪性。

嵌入式操作系统发展经历了从支持 8 位微处理器到支持 16 位、32 位甚至 64 位微处理器, 以及从支持单一品种的微处理器芯片到支持多品种微处理器芯片、从只有内核到除了内核外还提供其他功能模块(如文件系统、TCP/IP 网络系统、窗口图形系统)等过程。

典型的嵌入式操作系统有:

μC/OS-II、μClinux、VxWorks、Windows CE、Symbian OS、Android、iPhone OS 等。

μC/OS-II 能管理 64 个任务, 其实时性能优良和可扩展性好, 可运行在航天器等对安全级别要求较高的系统上。

μClinux 继承了 Linux 操作系统的主要特性, 内核非常小, 有良好的稳定性和移植性、强大的网络功能、出色的文件系统支持、标准丰富的 API 以及 TCP/IP 网络协议等。

VxWorks 以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中。

Windows CE 是微软自行开发的嵌入式新型操作系统, 具有模块化、结构化和基于 Win32 应用程序接口和与处理器无关等特点。

Symbian OS 是智能移动终端的专用嵌入式操作系统, 它可支持 Java 语言, 拥有强大的应用程序及通信处理开发能力。

Android 是 Google 开发的基于 Linux 平台的开源移动终端操作系统, 它采用 WebKit 浏览器引擎, 具备触摸屏、高级图形显示和上网功能。Android 应用开发是基于 Java 的, 底层是基于 Linux 的。开发者在其上开发应用程序自由度大, 而且系统可免费获得, 已成为流行的嵌入式操作系统。

iPhone OS 是苹果公司为 iPhone、iPad 开发的操作系统, 它有很好的媒体处理与触屏交互处理功能支持。

1.4 操作系统的运行环境

我们知道, 一个程序在计算机上运行是需要具有一定的环境的。例如, 要有处理器、主