高等教育规划教材

操作系统原理及应用(Linux)

汪杭军　主编
楼吉林　张镇潮　崔坤鹏　副主编

免费提供电子教案

下载网址 http://www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

高等教育规划教材

操作系统原理及应用（Linux）

汪杭军　主编
楼吉林　张镇潮　崔坤鹏　副主编

机械工业出版社

本书讲述了操作系统的基本原理、概念和应用，涵盖了操作系统概论、进程管理、内存管理、设备管理和文件管理；同时以 Linux 系统为主线，对 Fedora 系统安装、桌面系统的使用、Linux 应用程序的安装和升级、服务器环境配置、Linux 环境下的 C 语言编程，以及 Linux 内核构建等实践内容进行了介绍；最后，以桌面虚拟化管理为例分析了 Linux 的具体应用案例。

本书既可作为高等学校计算机相关专业本、专科的教材，也可作为非计算机专业人员深入学习操作系统理论和实践知识的教材和辅导书，同时也适合作为广大学生自学和考研复习的参考书使用。

本书配有授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ：2850823885，电话：010－88379739）。

图书在版编目(CIP)数据

操作系统原理及应用：Linux/汪杭军主编．—北京：机械工业出版社，2016.9
高等教育规划教材
ISBN 978-7-111-54961-1

Ⅰ．①操… Ⅱ．①汪… 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（2016）第 232787 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码　100037）
策划编辑：郝建伟　　责任编辑：郝建伟
责任校对：张艳霞
责任印制：李　洋

中国农业出版社印刷厂印刷

2017 年 1 月第 1 版·第 1 次印刷
184 mm×260 mm·14.75 印张·279 千字
0001－3000 册
标准书号：ISBN 978-7-111-54961-1
定价：39.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
服务咨询热线：(010)88379833
读者购书热线：(010)88379649

网络服务
机 工 官 网：www.cmpbook.com
机 工 官 博：weibo.com/cmp1952
教育服务网：www.cmpedu.com
金　书　网：www.golden－book.com

出 版 说 明

当前，我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才，是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才，但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势，配合高等院校的教学改革，满足高质量高校教材的迫切需求，机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门，通过充分的调研和讨论，针对相关课程的特点，总结教学中的实践经验，组织出版了这套“高等教育规划教材”。

本套教材具有以下特点：

1）符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，加大案例篇幅或实训内容，强调知识、能力与素质的综合训练。

2）针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。

3）凝结一线骨干教师的课程改革和教学研究成果，融合先进的教学理念，在教学内容和方法上做出创新。

4）为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5）注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助！

机械工业出版社

前　言

操作系统是计算机系统的基本组成部分，是整个计算机系统的基础和核心。正是由于操作系统的重要地位，它已成为各大专院校计算机相关专业的一门必修课程。但是，操作系统课程本身的概念较多、内容抽象难懂，初学者要掌握它需要花费很大的心思。而作为教材，如何合理编排教学内容，将操作系统的原理和实践应用结合起来，使学习者能够融会贯通，从而在工作和生活中发挥操作系统的作用，并能够真正解决问题，这是值得人们不断努力去探讨的一件事。

在很多院校中，尤其是独立学院和高职高专院校，其操作系统的教学偏重于理论部分，而采用的大部分教材主要也是阐述操作系统的概念和原理。这些内容偏难、过于抽象，如进程管理、内存管理等，大多需要学生去想象，如果没有一个良好的编程基础，根本无从理解。与一些重点院校不同，这些院校的大部分同学对深入操作系统内部的需求不大，往往只是需要比较方便地理解操作系统的基本原理，然后能够对 Linux 操作系统的应用有更多的要求。虽然现有的一些教材中加入了关于 Linux、UNIX 或 Windows 系统的介绍，但是它们大多还是其前面理论部分的重复和延伸，或者是加入实际操作系统的源码理解，很难满足这部分大专院校和很多操作系统初学者的需求。

本书内容本着重基础、重能力、求创新、突出职业应用的总体思想，结合创新创业型高等院校的教学要求和 IT 职业的能力需求，并兼顾硕士研究生入学考试知识点，经专家组多次讨论审订修改确定。

本书主体内容基于浙江农林大学和浙江省绍兴市的《操作系统》精品课程建设，通过十几年来操作系统的教学和项目指导，在编者积累经验和资料的基础上最终整理而成。本书从实用的角度出发，充分考虑了学习者对于操作系统原理和实践应用所需要掌握的知识，内容包括：第 1 章引言，包含计算机系统的主要组成部分和原理概述，以及操作系统的概念、发展及特征等内容；第 2 章进程管理，介绍了进程的概念、状态、描述和控制、互斥和同步，以及处理器调度、线程和死锁等知识；第 3 章内存管理，介绍了分区管理、页式、段式和段页式管理方式，并讨论了虚拟存储技术；第 4 章设备管理，介绍了 I/O 的组织、设计、缓冲，以及磁盘调度、RAID 和磁盘高速缓存；第 5 章文件管理，介绍了文件的相关概念、组织结构与存取方式，文件目录管理，存储空间管理，以及文件的共享和保护问题；第 6 章 Fedora 操作系统，介绍了 Fedora 操作系统及其安装；第 7 章 Fedora 桌面系统的使用，介绍了桌面系统的常规使用、网络配置和常用命令行；第 8 章 Linux 应用程序的安装和管理，介绍了安装 Linux 系统的几种方法，包括 yum、RPM 包和源代码安装应用的问题；第 9 章 Linux 服务器环境配置，介绍了 Java、Tomcat、MySQL、Apache 和 PHP 的环境安装与配置；第 10 章 Linux 环境下 C 语言编程基础，介绍了编程工具 vi、gcc 和 gdb 的使用，以及程序查错和调试的方法；第 11 章构建 Linux 内核，介绍了如何从源代码开始配置和编译 Linux 内核，以及引导加载设置；第 12 章以桌面虚拟化管理为例，介绍了 Linux 虚拟化技术，以及通过 oVirt 虚拟化管理平台的应用。全书深浅适度，安排系统、合理。

本书包括了操作系统的实践应用的各个方面，实用性很强，可作为高等学校计算机相关专业本、专科教材，也可作为非计算机专业的人员深入学习操作系统理论和实践知识的教材和辅导书，同时也适合广大学生自学和考研复习使用，另外，对于 Linux 系统和网络管理人员而言，本书也是一本很好的参考书。

本书计划讲课学时为 72 学时，不同的学校和专业可根据需要删去或略讲书中的某些章节。

本书第 1、2、8、10 章由汪杭军编写，第 3、4、6、9 章由楼吉林编写，第 5、7 章由崔坤鹏编写，第 11、12 章由张镇潮和张八一编写，全书由汪杭军统稿。

由于时间仓促，加上作者水平有限，教学需要不断更新完善，书中难免存在一些错误或不妥之处，恳请广大读者谅解。也欢迎对本书内容提出批评和修改建议，对此将不胜感激。如有需要请联系编者（Email:whj@ zafu. edu. cn）。

编　者

目　录

出版说明

前言

第 1 章　引言：计算机系统和操作系统概述 …… *1*

1.1　计算机系统概述 …… *1*

1.1.1　计算机的基本组成 …… *1*

1.1.2　处理器寄存器和指令执行 …… *2*

1.1.3　中断 …… *6*

1.1.4　存储器 …… *8*

1.1.5　I/O 访问方式 …… *9*

1.2　操作系统概述 …… *11*

1.2.1　操作系统的概念及功能 …… *11*

1.2.2　操作系统的发展 …… *12*

1.2.3　操作系统的结构 …… *17*

1.2.4　现代操作系统的基本特征 …… *18*

1.3　思考与练习 …… *19*

第 2 章　进程管理 …… *21*

2.1　进程的概念及其特性 …… *21*

2.1.1　进程的定义 …… *21*

2.1.2　进程的特性 …… *21*

2.2　进程状态 …… *22*

2.2.1　两状态进程模型 …… *22*

2.2.2　五状态进程模型 …… *23*

2.2.3　挂起进程模型 …… *24*

2.3　进程描述和控制 …… *25*

2.3.1　进程描述内容 …… *25*

2.3.2　执行模式 …… *26*

2.3.3　进程控制操作 …… *26*

2.3.4　进程切换 …… *27*

2.4　进程互斥和同步 …… *27*

2.4.1　进程交互方式 …… *28*

2.4.2　进程互斥要求 …… *28*

2.4.3　进程互斥的实现 …… *28*

2.4.4　信号量实现进程的同步与互斥 …… *30*

2.4.5　管程和消息传递 …… *34*

2.5　处理器调度 …… *34*

2.5.1　处理器调度的类型 …… *34*

2.5.2　调度的衡量标准 …… *35*

2.5.3　处理器调度算法 …… *36*

2.6　线程 …… *38*

2.6.1　线程的基本概念 …… *38*

2.6.2　线程管理实现机制 …… *39*

2.6.3　多线程的应用 …… *40*

2.7　死锁 …… *42*

2.7.1　死锁的原理 …… *42*

2.7.2　死锁预防 …… *43*

2.7.3　死锁避免 …… *44*

2.7.4　死锁检测和恢复 …… *47*

2.8　思考与练习 …… *47*

第 3 章　内存管理 …… *50*

3.1　计算机存储结构 …… *50*

3.1.1　存储器配置方式 …… *50*

3.1.2　常见 PC 存储结构 …… *52*

3.2　地址重定位及内存访问保护 …… *54*

3.2.1　地址空间 …… *54*

3.2.2　地址重定位 …… *54*

3.2.3　地址重定位及存储信息保护 …… *57*

3.3　分区存储管理技术 …… *59*

3.3.1　单一分区内存管理 …… *59*

3.3.2　固定大小的多分区管理 …… *60*

3.3.3　动态分区管理 …… *61*

3.4　分区分配算法 …… *65*

3.4.1　分区分配算法描述 …… *65*

3.4.2 分配算法使用特性 …… 67
3.5 页式管理 …… 68
3.5.1 分页的基本思想 …… 68
3.5.2 静态页式管理 …… 68
3.5.3 动态页式管理 …… 69
3.6 段式管理 …… 70
3.6.1 段式管理的基本原理 …… 70
3.6.2 地址变换机构 …… 71
3.7 段页式管理 …… 71
3.7.1 分页与分段管理的特点 …… 71
3.7.2 段页式管理方式 …… 72
3.8 虚拟存储技术 …… 74
3.8.1 局部性原理 …… 75
3.8.2 虚拟存储的基础 …… 75
3.8.3 用分页管理实现虚拟存储 …… 76
3.8.4 虚拟存储页面置换算法 …… 78
3.9 思考与练习 …… 81
第4章 设备管理 …… 82
4.1 I/O 设备功能的组织 …… 82
4.1.1 I/O 功能的发展 …… 82
4.1.2 直接存储器访问 …… 84
4.2 操作系统设计问题 …… 87
4.2.1 设计目标 …… 87
4.2.2 I/O 功能的逻辑结构 …… 88
4.3 I/O 缓冲 …… 91
4.3.1 单缓冲 …… 91
4.3.2 双缓冲 …… 92
4.3.3 循环缓冲 …… 92
4.3.4 缓冲的作用 …… 93
4.4 磁盘调序 …… 94
4.4.1 磁盘性能参数 …… 94
4.4.2 磁盘调度策略 …… 95
4.5 RAID …… 97
4.5.1 RAID 0 …… 98
4.5.2 RAID 1 …… 99
4.5.3 RAID 2 …… 100
4.5.4 RAID 3 …… 101
4.5.5 RAID 4 …… 102
4.5.6 RAID 5 …… 102
4.5.7 RAID 6 …… 102
4.6 磁盘高速缓存 …… 103
4.6.1 设计考虑 …… 103
4.6.2 性能考虑 …… 104
4.7 思考与练习 …… 105
第5章 文件管理 …… 106
5.1 文件管理概述 …… 106
5.1.1 文件和文件系统 …… 106
5.1.2 文件管理的功能 …… 106
5.1.3 文件管理系统的层次结构 …… 107
5.2 文件的组织结构与存取方式 …… 108
5.2.1 堆文件 …… 109
5.2.2 顺序文件 …… 110
5.2.3 索引顺序文件 …… 110
5.2.4 索引文件 …… 111
5.2.5 直接文件或散列文件 …… 112
5.3 文件目录管理 …… 112
5.3.1 文件目录 …… 112
5.3.2 文件目录结构 …… 113
5.3.3 文件控制块 …… 115
5.3.4 目录与文件 …… 115
5.4 存储空间管理 …… 116
5.4.1 空闲块表法 …… 116
5.4.2 空闲块链法 …… 116
5.4.3 位示图法 …… 117
5.4.4 成组链接法 …… 118
5.5 文件共享与文件保护 …… 118
5.5.1 文件共享方法 …… 118
5.5.2 文件保护方式 …… 121
5.6 思考与练习 …… 122
第6章 Fedora 操作系统 …… 123
6.1 Fedora 操作系统简介 …… 123
6.2 Fedora 操作系统的安装 …… 126
6.2.1 基本设置 …… 126
6.2.2 磁盘分区及软件包选择 …… 129
6.2.3 最终设置 …… 132
6.3 思考与练习 …… 134
第7章 Fedora 桌面系统的使用 …… 135

7.1 登录、注销与关机 …… 135
7.1.1 开机与登录 …… 135
7.1.2 锁屏、注销与关机 …… 136
7.2 使用 GNOME 桌面 …… 138
7.2.1 查看 GNOME 桌面系统版本 … 138
7.2.2 使用 GNOME 桌面工具管理 Linux …… 138
7.3 Fedora 网络配置 …… 143
7.4 使用命令行 …… 145
7.4.1 认识命令行 …… 145
7.4.2 命令的语法 …… 146
7.4.3 常用命令 …… 147
7.5 思考与练习 …… 150
第 8 章 Linux 应用程序的安装和管理 …… 151
8.1 使用 yum 命令安装和升级应用程序 …… 151
8.1.1 在线安装 …… 151
8.1.2 本地安装 …… 153
8.1.3 其他功能 …… 155
8.2 管理 RPM 软件包 …… 156
8.3 从源代码安装应用程序 …… 159
8.3.1 准备工作 …… 159
8.3.2 使用源代码进行安装 …… 159
8.4 把应用程序的图标添加到桌面上 …… 163
8.5 常用应用程序推荐列表 …… 165
8.6 思考与练习 …… 167
第 9 章 Linux 服务器环境配置 …… 168
9.1 Java 开发环境的安装与配置 … 168
9.2 Tomcat 服务器的安装与配置 …… 170
9.3 MySQL 数据库的安装与配置 …… 174
9.4 Apache 服务器的安装与配置 …… 176
9.5 PHP 环境的安装与配置 …… 178
9.6 思考与练习 …… 180
第 10 章 Linux 环境下 C 语言编程基础 …… 181
10.1 准备知识 …… 181
10.1.1 vi 编辑器 …… 181
10.1.2 gcc 编译器和 gdb 调试器 …… 182
10.2 Linux 简单 C 程序实现 …… 184
10.3 程序查错及调试 …… 193
10.4 思考与练习 …… 198
第 11 章 构建 Linux 内核 …… 201
11.1 下载、安装和预备内核源代码 …… 201
11.1.1 相关信息和先决条件 …… 201
11.1.2 下载和安装源代码 …… 202
11.2 配置和编译 Linux 内核 …… 204
11.2.1 配置内核 …… 204
11.2.2 定制内核 …… 206
11.2.3 编译 …… 207
11.3 安装内核、模块和相关文件 …… 208
11.4 GRUB：Linux 引导加载程序 …… 208
11.5 思考与练习 …… 209
第 12 章 Linux 应用案例（桌面云） …… 210
12.1 云的概念和桌面虚拟化 …… 210
12.2 基于 Linux 的虚拟化技术 …… 210
12.2.1 Xen 技术 …… 210
12.2.2 KVM 技术 …… 212
12.3 oVirt 虚拟化管理平台 …… 213
12.3.1 oVirt 架构和运行基础 …… 213
12.3.2 基于 CentOS 7 的环境准备 … 217
12.3.3 ovirt - engine 安装 …… 219
12.3.4 ovirt - note 安装 …… 220
12.3.5 操作系统设置 …… 220
12.3.6 oVirt 配置 …… 220
12.3.7 虚拟机的创建和管理 …… 221
12.3.8 大规模部署虚拟机 …… 227
参考文献 …… 228

第1章　引言：计算机系统和操作系统概述

本章的目的是为本书的其他部分提供操作系统的相关背景知识，包括计算机系统结构和操作系统核心中的基本概念。

第一节计算机系统概述是对计算机系统中的处理器、中断、存储器和输入/输出的简要介绍。为了更好地理解操作系统的功能及原理，掌握计算机组织与系统结构是非常重要的，这是由操作系统的地位决定的：它的一边是应用程序、实用程序和用户，而另一边则是计算机硬件。

第二节操作系统概述是关于操作系统的一个简要大纲，以便初学者能对操作系统涉及的众多领域有一个粗略的了解，包括操作系统的概念和功能，操作系统的历史发展，操作系统采用的体系结构，以及现代操作系统的重要特征。

1.1　计算机系统概述

本书假定读者已经学习过计算机系统硬件的相关知识。本节将对操作系统中，特别是本书后面内容相关的计算机系统硬件内容进行简要概述。掌握这些底层的计算机系统硬件知识对于理解操作系统的原理、功能和设计是很重要的。

1.1.1　计算机的基本组成

大家知道，一个完整的计算机系统包括硬件系统和软件系统两大部分。这里所讨论的是计算机硬件系统。通常，人们把不装备任何软件的计算机称为硬件计算机或裸机。

计算机硬件的基本功能是接受计算机程序的控制来实现数据输入、运算和数据输出等一系列根本性的操作。目前计算机的基本硬件结构一直沿袭着传统的冯·诺伊曼框架，即由运算器、控制器、存储器、输入设备和输出设备五大部件构成。

1. 运算器

运算器是计算机中执行各种算术和逻辑运算操作的部件，其基本操作包括加、减、乘、除四则运算，与、或、非、异或等逻辑操作，以及移位、比较和传送等操作，也称算术逻辑单元（Arithmetic Logic Unit，ALU）。

2. 控制器

控制器是指挥计算机的各个部件按照指令的功能要求协调工作的部件，它本身不具有运算功能，而是通过读取各种指令，并对其进行翻译和分析，而后对各部件做出相应的控制。它主要由指令寄存器、译码器、程序计数器和操作控制器等组成。

3. 存储器

存储器是计算机的记忆和存储部件，用来存放信息，包括程序和数据，并根据控制命令提供这些数据和程序。计算机存储器可分为两部分：一个是包含在计算机主机中的内存储器

(或称为内存、主存)，它直接和运算器、控制器交换数据，用于存放正在处理的数据或正在运行的程序；另一个是外存储器，又称为辅助存储器，它间接和运算器、控制器交换数据，用来存放暂时不用的数据。

4. 输入/输出设备

输入设备是外界向计算机传送信息的装置，它负责把用户的信息（包括程序和数据）输入到计算机中；而输出设备负责将计算机中的信息（包括程序和数据）传送到外部媒介，并转化成某种为人们所认识的表示形式，供用户查看或保存。常用的输入设备有键盘、鼠标、扫描仪、音频输入设备和视频输入设备等，常用的输出设备有显示器、打印机等。

📖 中央处理器，或简称处理器（Central Processing Unit，CPU），是计算机系统的核心，由运算器和控制器两个部件组成。CPU 是计算机的心脏，其品质的高低直接决定了计算机系统的档次。CPU 的主要性能指标有两个：字长和主频。

系统总线（System Bus）用来连接计算机各功能部件（包括处理器、内存和输入/输出模块），使它们构成一个完整的微机系统。系统总线上传送的信息包括数据信息、地址信息和控制信息，因此，系统总线包含三种功能的总线，即数据总线 DB（Data Bus）、地址总线 AB（Address Bus）和控制总线 CB（Control Bus）。

1.1.2 处理器寄存器和指令执行

1. 寄存器

寄存器（Register）是中央处理器内拥有有限存储容量的高速存储部件，可用来暂存指令、数据和地址。一般在 CPU 中至少包含六类寄存器：数据寄存器（DR）、指令寄存器（IR）、程序计数器（PC）、地址寄存器（AR）、累加寄存器（AC）和程序状态字寄存器（PSW）。

（1）数据寄存器

数据寄存器（Data Register，DR），又称存储器缓冲寄存器（MBR），其主要功能是作为 CPU 和主存、外设之间信息传输的中转站，用以弥补 CPU 和主存、外设之间操作速度上的差异。

数据寄存器用来暂时存放由主存储器读出的一条指令或一个数据字；反之，当向主存存入一条指令或一个数据字时，也将它们暂时存放在数据寄存器中。

（2）指令寄存器

指令寄存器（Instruction Register，IR）用来保存当前正在执行的一条指令。当执行一条指令时，首先把该指令从主存读取到数据寄存器中，然后再传送至指令寄存器。

（3）程序计数器

程序计数器（Program Counter，PC）用来指出下一条指令在主存中的地址。在程序执行之前，首先必须将程序的首地址，即程序第一条指令所在主存单元的地址送入 PC，因此 PC 的内容即是从主存提取的第一条指令的地址。

当执行指令时，CPU 能自动递增 PC 的内容，使其始终保存将要执行的下一条指令的主存地址，为读取下一条指令做好准备。

但是，当遇到转移指令时，下一条指令的地址将由转移指令的地址码字段来指定，而不是像通常那样通过顺序递增 PC 的内容来获得。

（4）地址寄存器

地址寄存器（Address Register,AR），又称存储器地址寄存器（MAR），用来保存 CPU 当前所访问的主存单元的地址。由于在主存和 CPU 之间存在操作速度上的差异，所以必须使用地址寄存器来暂时保存主存的地址信息，直到主存的存取操作完成为止。

当 CPU 和主存进行信息交换，即 CPU 向主存存入数据/指令或者从主存读出数据/指令时，都要使用地址寄存器和数据寄存器。

同样，人们通常也把外围设备与主存单元一样进行统一编址，当 CPU 和外围设备交换信息时，同样也需要使用地址寄存器和数据寄存器。

（5）累加寄存器

累加寄存器通常简称累加器（Accumulator,AC），是一个通用寄存器。其功能是：当运算器的算术逻辑单元 ALU 执行算术或逻辑运算时，为 ALU 提供一个工作区，可以为 ALU 暂时保存一个操作数或运算结果。

（6）程序状态字寄存器

程序状态字（Program Status Word,PSW）用来标识当前 CPU 的运算状态及程序的工作方式。

程序状态字寄存器用来保存由算术/逻辑指令运行或测试的结果所建立起来的各种条件码内容，如运算结果进/借位标志（C）、运算结果溢出标志（O）、运算结果为零标志（Z）、运算结果为负标志（N）和运算结果符号标志（S）等，这些标志位通常用 1 位触发器来保存。除此之外，程序状态字寄存器还用来保存中断和系统工作状态等信息，以便 CPU 和系统及时了解计算机运行状态和程序运行状态。

2. 指令执行

计算机的任何行动都需要依赖程序的指挥来完成，而程序最终都会转化成一条条能被处理器执行的指令。接下来将介绍这些指令是如何被处理器执行的。正确执行这些指令需要依赖于前面介绍的处理器内部的各种寄存器。

CPU 处理一条指令最简单的方式由两个步骤组成：处理器从存储器中读取一条指令，然后执行这条指令。这两个步骤所需的全部时间称为指令周期，由取指周期和执行周期组成。程序的执行就是不断重复取指令和执行指令的过程。不同的指令，其执行涉及的操作会有很大的不同，而且每个指令的指令周期也会有很大的差异。图 1-1 给出了一个指令周期的流程图示意，其中中断周期部分将在下一小节中进行描述。仅当机器关机、发生某些未发现的错误或者遇到与停机相关的程序指令时，程序执行才会停止。

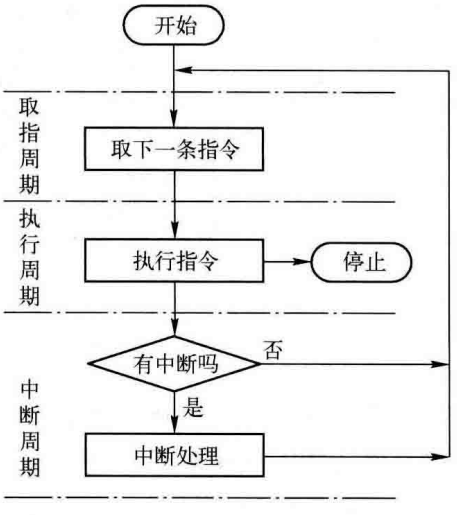


图 1-1　指令周期

指令包括操作码和地址码两个字段，其中操作码表示指令的操作特性与功能，并通过指令译码器（Instruction Decoder，ID）对操作码进行译码，产生指令操作所需的控制电位，并将其送到微操作控制线路上，在时序部件定时信号的作用下，产生具体的操作控制信号；地址码字段通常指定参与操

作的操作数的地址。

指令的操作可分为以下 4 种类型。

1）处理器 - 存储器：数据可以从处理器传送到存储器，或者从存储器传送到处理器。

2）处理器 - I/O：数据可以输出到外部设备，或者从外部设备输入数据。

3）数据处理：执行与数据相关的算术操作或逻辑操作。

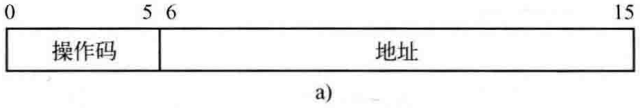
4）控制：转移操作，改变执行顺序，跳转到相应的指令地址。

然而，一条指令的执行可能涉及这 4 种操作的若干组合。不同的指令用操作码字段的不同编码来表示，每一种编码代表一种指令。组成操作码字段的位数一般取决于计算机指令系统的规模。例如，一个指令系统只有 8 条指令，则有 3 位操作码足够；如果有 32 条指令，那么就需要 5 位操作码。

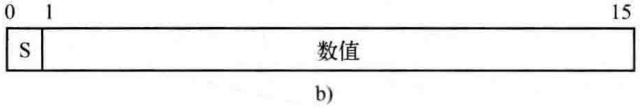
在每个指令周期开始时，处理器根据程序计数器（Program Counter，PC）的指令地址，从存储器中读取一条指令，放置在处理器的指令寄存器（Instruction Register，IR）中。然后，处理器在一般情况下递增 PC，使得它能够按顺序取得下一条指令（即位于下一个存储器地址的指令）。

下面通过一个实例（将两个内存单元中的数相加后存入其中一个单元中）来具体说明指令是如何在处理器中执行的。

考虑一台 16 位简化计算机，其指令和数据均为 16 位。其中 16 位的指令格式中前 6 位是操作码，后 10 位是地址。这样这台计算机最多可表示 $2^6 = 64$ 种不同操作；可直接访问的存储器最大为 $2^{10} = 1024 = 1\ \mathrm{K}$。该计算机的特征如图 1-2 所示。



a)



b)

序号	操作码（二进制）	功能
1	000001	从存储器中加载AC
2	000010	把AC的内容存储到存储器中
3	000011	从存储器中加到AC中
4	000100	从AC中减去指定字的内容
5	000101	从I/O中载入AC
6	000110	把AC保存到I/O中

c)

图 1-2　一台简化计算机的特征

a）指令格式　b）整数格式　c）操作码列表（部分）

程序实现 1975 与 2004 两个数相加，这两个数首先已分别存入地址为 398H 和 399H 的存储器单元中（转换为十六进制分别为 07B7 和 07D4），最后将运算结果存入地址 399H 中。该程序由 3 条指令 0798H、0F99 和 0B99 组成，并存入地址为 1F4H 开始的 3 个存储单元中。假设程序计数器 PC 初值为地址 1F4H，即处理器将在地址为 1F4H 的存储单元处读取指令，在随后的指令周期中，它将从地址为 1F5H、1F6H 等存储单元处读取指令。计算机初始状态如图 1-3 所示，图中的数据均以十六进制表示。

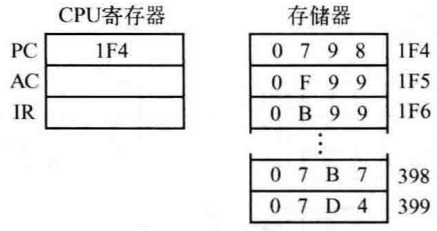


图 1-3　计算机初始状态

图 1-4 显示了执行第一条指令的状态。取指阶段，由 PC 指出的存储器 1F4 中取一条指令 0798 到 IR 寄存器中，然后 PC 自动增 1；执行阶段，对 IR 中的指令进行译码，得到 0000011110011000B，前 6 位为操作码 000001B，查操作码列表得到为“从存储器中加载 AC”，存储器地址由后面的 10 位确定，即 1110011000B = 398H。于是执行该指令后，AC 中的值被设置为从 398H 存储器中获得的 07B7H。

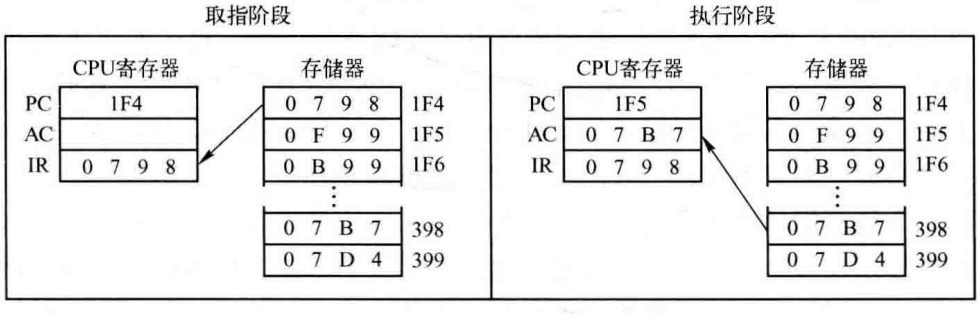


图 1-4　第一条指令执行状态

图 1-5 显示了执行第二条指令的状态。取指阶段，从 1F5 中取出指令 0F99 到 IR 中，然后 PC 增 1；执行阶段，指令译码为 0000111110011001B，操作码为 000011B，即为“从存储器中加到 AC 中”，其中存储器地址为后 10 位，即 1110011001B = 399H。于是该指令执行为：将 AC 中的值（07B7）与存储器 399H 中内容（07D4）相加后再存入 AC 中（0F8B）。

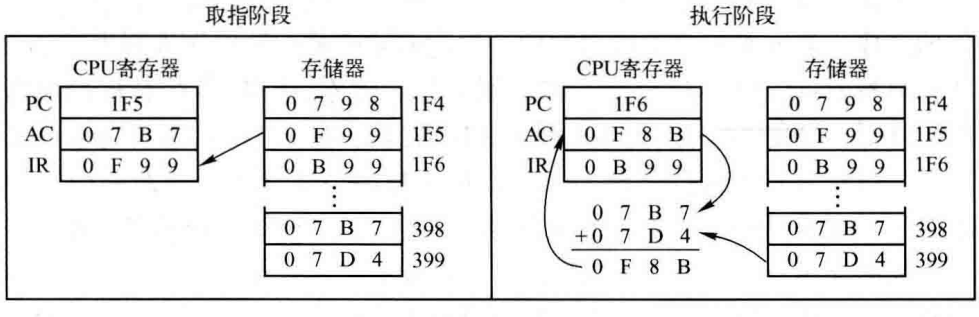


图 1-5　第二条指令执行状态

图 1-6 显示了执行第三条指令的状态。取指阶段，从 1F6 中取出指令 0B99 到 IR 中，然后 PC 增 1；执行阶段，指令译码为 0000101110011001B，操作码为 000010B，即“把 AC 的内容存储到存储器中”，其中存储器地址为后 10 位，即 1110011001B = 399H。于是该指令执行后，将 AC 中的值（0F8B）存储到 399H 的存储器单元中。

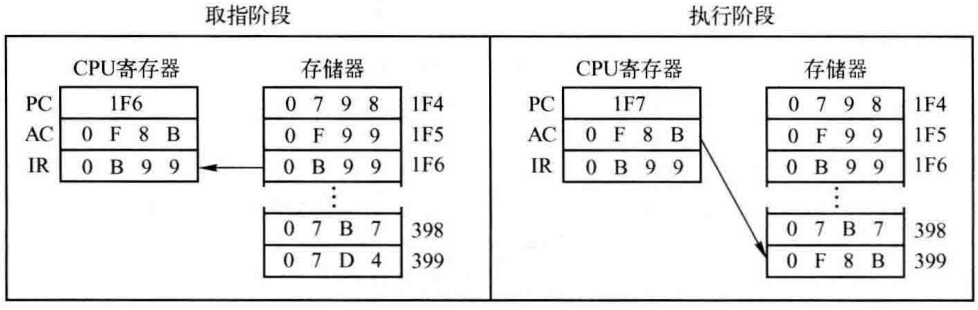


图 1-6　第三条指令执行状态

1.1.3　中断

中断是计算机中的一个十分重要的概念，在操作系统中都要采用中断技术。对于一个运行在计算机上的操作系统而言，缺少了中断机制，将是不可想象的。

中断（Interrupt）是指处理器接收到来自硬件或软件的信号，提示发生了某个事件，应该立即去处理。根据信号来源的不同，中断分硬件中断(信号来自硬件）和软件中断(信号来自软件）两种。常见的硬件中断有时钟中断、I/O 中断和硬件故障中断等；软件中断通常是一些程序性事故，如算术溢出、非法操作码、地址越界和除法非法等。

中断机制为计算机的硬件设备和软件提供了一种交流的途径。举一个日常生活中的例子来说明。假如你正在看一本书，这时手机来电了，你放下书，去接电话；通完电话后，从刚才的内容继续接着阅读。这里手机发出的信号可看作“中断请求”，它使你暂时中止当前的工作（阅读），而去处理更为急需处理的事情（接电话），这个过程可看作“中断响应”，而接电话的过程就是“中断处理”；把急需处理的事情处理完毕，再回头来继续做原来的事情。

然而中断最初的目的是为了提高处理器效率。再来看一个例子，假如有客人某天要来拜访你，但不确定具体时间，而且客人也不知你所在的具体位置。这样你只能在小区大门或在马路边等待，于是这段时间你什么事情也干不了。有了手机后，你就不必在门口等待而是可以去做其他的工作，客人来了打手机通知你。你这时才中断你的工作去到指定地点，这样就避免了等待和浪费时间。计算机也是一样，例如打印输出，CPU 传送数据的速度高，而打印机打印的速度低，如果不采用中断技术，CPU 将经常处于等待状态，效率极低。而采用了中断方式后，CPU 就可以进行其他的工作，只在打印机缓冲区中的当前内容打印完毕发出中断请求之后，才予以响应，暂时中断当前工作转去执行向缓冲区传送数据，传送完成后又返回执行原来的程序。这样就极大地提高了处理器的工作效率。

利用中断，处理器可以在 I/O 操作的执行过程中执行其他指令。这种 I/O 操作和用户程序的执行是并发的。当外部设备完成 I/O 操作后，给处理器发送一个中断请求信号。处理器做出中断响应，暂停当前程序的处理，转去处理 I/O 设备程序，称为中断处理程序（interrupt handler）。在对该设备的服务响应处理完成后，处理器恢复原程序的执行。从用户程序的角度看，中断虽然打断了程序的正常执行，但是中断处理完成后，又恢复了执行。这个中断及中断的处理过程示例如图 1-7 所示，在处理器执行用户程序指令 i 处发生中断，经过以下一些过程。

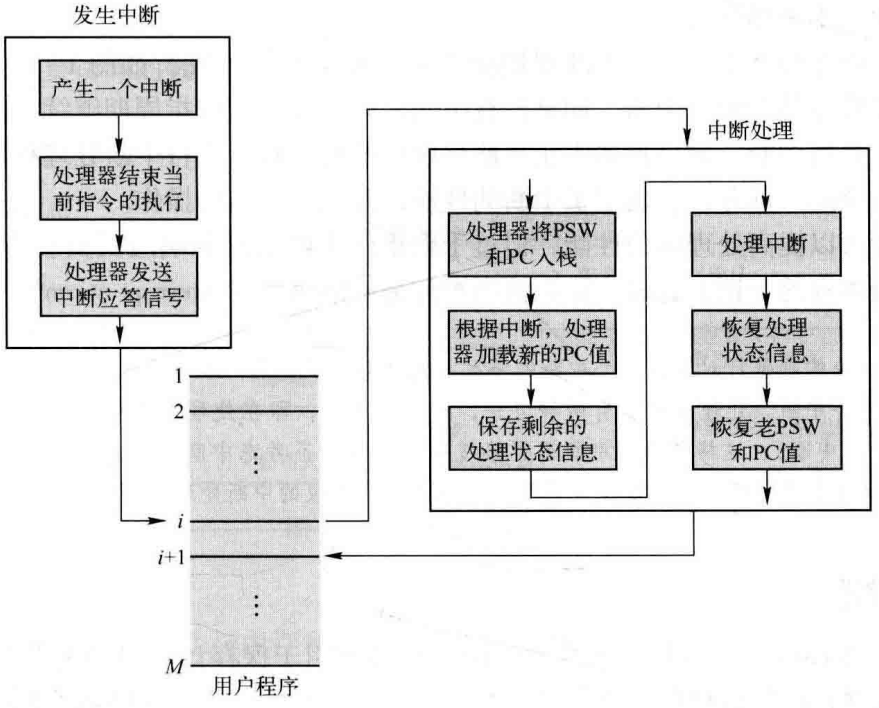


图 1-7　中断转移及中断处理

1）设备给处理器发出一个中断信号。

2）处理器结束当前指令的执行，然后响应中断。

3）处理器对中断进行确认，并向该设备发送中断应答信号。

4）处理器在把控制权转移到中断程序前需要保护现场，即保存断点和寄存器信息。至少包括程序状态字 PSW 和程序计数器 PC，它们被压入系统栈（内存中属于操作系统空间的一块区域，用于保存中断现场和操作系统子程序间相互调用的参数、返回值、返回点及子程序的局部变量）。

5）处理器把响应此中断的中断处理程序入口地址装入程序计数器 PC 中，并进入下一个指令周期，即控制被转移到中断处理程序。中断处理程序继续执行以下操作。

6）保存其他有关正在执行程序的状态信息，特别是处理器寄存器内容（中断处理程序可能会用到这些寄存器，从而破坏这些寄存器原有的内容，会引起返回断点后原程序出错）。其他还必须保存的状态信息将在第 2 章的进程内容中进行描述。

7）中断处理程序开始处理中断，其中包括检查与 I/O 操作相关的状态信息或其他引起中断的事件，还有可能包括给 I/O 设备发送附加命令或应答等。

8）当中断处理结束后，被保存的寄存器值等状态信息从系统栈中释放并恢复。

9）从系统栈中恢复 PSW 和 PC，下一条要执行的指令来自被中断的程序，从而使中断处理程序完成后，处理器在中断点恢复对用户程序的执行。

从中断的处理过程可以看到，用户程序并不需要为中断添加任何特殊的代码，处理器和操作系统负责挂起用户程序，然后在同一个地方恢复执行。为了保证程序中断后还能正确运行，保存被中断程序的所有状态信息并在以后恢复这些信息是十分重要的。这是由于中断并不是程序调用的一个例程，它可以在任何时候发生，在用户程序执行过程中的任何一点上发

生，它的发生是不可预测的。

因此，在指令周期中有一个中断周期阶段来适应中断的处理，如图 1-1 所示。在中断周期，处理器检查是否发生中断。如果没有中断，处理器转至取指周期继续运行当前程序的下一条指令；若有中断，则处理器中止当前程序的执行，转为执行中断处理程序。中断处理程序是操作系统的一部分，它确定了中断的性质，并完成所需的操作。

中断尽管可以提高处理器的性能，但过于密集的中断请求和响应反而会影响系统性能，这是由于中断本身的开销引起的。这类情形被称为中断风暴（interrupt storm）。

到这里，所讨论的都是针对发生一个中断的情况。假设存在多中断，即当处理一个中断时，可以发生另外一个或者多个中断。处理多中断有两种方法，一是禁中断，即在处理一个中断时，禁止再发生中断（挂起新发生的中断），这样所有的中断都严格按顺序处理，不考虑中断的相对优先级和时间限制的要求；二是定义中断优先级，允许高优先级的中断打断低优先级的中断处理程序的运行。

1.1.4 存储器

存储器（Memory）是计算机系统中的记忆设备，用于保存计算机中的各类信息，包括输入的原始数据、计算机程序、中间运行结果和最终运行结果等。存储器根据控制器指定的位置存入和取出信息。

由于存储器直接关系到计算机的程序和数据的保存，影响到计算机是否能正常工作，因此内存管理（Memory Management）和文件管理（File Management）都是操作系统设计中最重要、最复杂的内容。虽然计算机硬件一直在飞速发展，存储器的容量和存取速度都在不断增长，但是仍然不可能满足日益增长的用户程序和系统对存储的所有需求，所以需要了解各类存储部件，以及对这些部件的有效分配和管理，以获得更好的计算机性能。

1. 存储器分类

按用途存储器可分为主存储器（内存）和辅助存储器（外存）。外存通常是磁性介质或光盘等，能长期保存信息，属于外部设备。内存是主板上的存储部件，用来存放当前正在执行的数据和程序，负责直接与 CPU 交换指令和数据。广义上内存分为随机存储器（Random Access Memory，RAM）和只读存储器（Read Only Memory，ROM）。其中 RAM 用于暂时存放程序和数据，关闭电源或断电后，数据会丢失；ROM 主要用于存储计算机中重要的信息，如主板的 BIOS（基本输入/输出系统），其中保存的信息不会因为断电而丢失。为了高速地向 CPU 提供指令和数据，现在绝大多数的计算机都会配置高速缓冲存储器（Cache），从而加快了程序的执行速度。各类存储器如图 1-8 所示。

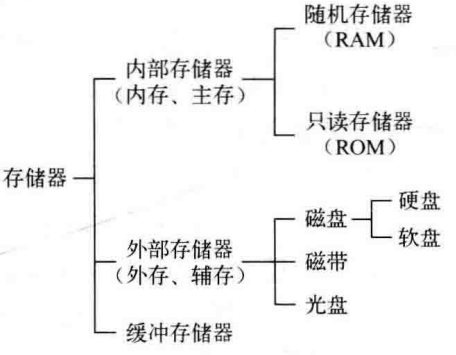


图 1-8　存储器分类

2. 存储器层次结构

不同的存储器，其价格、存储容量和存取时间是不一样的。越靠近 CPU 的存储，存取时间越短、存储容量越小、成本越高。在现有技术条件下，任何一种存储装置，都无法同时从时间、容量与价格这几个方面满足用户的需求。实际上它们组成了一个存取速度由快到