

高等学校 21 世纪教材

软件技术基础教程

徐士良 孙甲松 编著

人民邮电出版社

T/31-43
X7-5a1

图书在版编目(CIP)数据

软件技术基础教程 / 徐士良, 孙甲松编著. —北京: 人民邮电出版社, 2002. 12
高等学校 21 世纪教材
ISBN 7-115-10900-1

I. 软... II. ①徐...②孙... III. 软件—高等学校—教材 IV. TP31

中国版本图书馆 CIP 数据核字 (2002) 第 084438 号

内 容 简 介

本书是关于计算机软件技术的基础教材, 简明扼要地介绍了计算机软件设计与开发过程中所需要的基本知识。主要包括: 基本的数据结构及其运算, 实用的查找与排序技术, 数据库设计的技术, Windows 程序设计, 编译技术, 应用软件设计与开发技术等。本书从实用出发, 主要强调将系统软件的实现原理以及其中的基本技术用于应用软件的设计与开发。

本书是作者在长期教学实践的基础上编写成的, 内容既具有系统性, 又具有实用性, 叙述通俗易懂, 例题丰富。书中所有算法程序都经过实际调试, 正确有效。本书每章都附有大量的习题, 供学习过程中练习之用。

本书可以作为大专院校计算机软件教学的基本教材, 也可以作为各种培训班的教材以及自学教材。

高等学校 21 世纪教材 软件技术基础教程

◆ 编 著 徐士良 孙甲松
策划编辑 滑 玉
责任编辑 郭 玲

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67129260
北京汉魂图文设计有限公司制作
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16
印张: 23
字数: 557 千字 2002 年 12 月第 1 版
印数: 1-5 000 册 2002 年 12 月北京第 1 次印刷

ISBN 7-115-10900-1/TP · 3219

定价: 29.80 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

丛书前言

当今世界,科学技术突飞猛进,知识经济已见端倪,国际竞争日趋激烈。教育在综合国力的形成中处于基础地位,国力的强弱将越来越取决于劳动者的素质,取决于各类人才的质量和数量,这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初,我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体,是进行教学的基本工具,也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此,高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代,计算机教学内容也随之更新,尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施,对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授,精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略,组织编写好这套教材,我们在国内高校做了系统、详细的调查,对教育部制订的教育计划做了认真的研究,还对国内外已出版的教材做了理性的分析,确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中,我们多次组织了由专家及高校一线教师参加的研讨会,对现有比较出色的教材的特点及优点进行了分析,博采众长,力求实现教材权威性与实用性的完美结合。

本套教材有如下特点:

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况,在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。
2. 紧密结合当前技术的新发展,在阐述理论知识的同时侧重实用性。
3. 力求在概念和原理的讲述上严格、准确、精练,理论适中,实例丰富,写作风格上深入浅出,图文并茂,便于学生学习。
4. 为适应当前高校课程种类多、课时数要压缩的教学特点,教材不仅篇幅有很大的压缩,而且均配有电子教案,以满足现代教学新特点的需要,做到易教易学。
5. 所选作者均是国内有丰富教学实践经验的知名专家、教授,所编教材具有较高的权威性。

教育的改革不会停止,教材也会不断推陈出新。目前本套教材即将推出,将接受广大教学第一线教师的检验。

由于我们的水平和经验有限,这批教材在编审、出版工作中还存在不少缺点和不足,希望使用本套教材的学校师生和广大读者提出批评和建议,以便改进我们的工作,使教材质量不断提高。

目 录

第 1 章 基础知识	1
1.1 计算机软件概述	1
1.1.1 计算机软件及其分类	1
1.1.2 程序设计语言及其语言处理程序	2
1.2 操作系统的基本概念	4
1.2.1 操作系统的功能与任务	4
1.2.2 操作系统的发展过程	5
1.2.3 操作系统的分类	8
1.3 算法	11
1.3.1 算法的基本概念	11
1.3.2 算法描述语言	13
1.3.3 算法设计基本方法	15
1.3.4 算法的复杂度分析	25
习题	28
第 2 章 基本数据结构及其运算	29
2.1 数据结构的基本概念	29
2.2 线性表	36
2.2.1 线性表顺序存储结构	36
2.2.2 顺序表的插入与删除	38
2.3 栈及其应用	41
2.3.1 栈的基本概念	41
2.3.2 栈的顺序存储及其运算	43
2.3.3 栈的应用	45
2.4 队列及其应用	48
2.4.1 队列的基本概念	48
2.4.2 循环队列及其运算	49
2.4.3 队列的应用	52
2.5 线性链表	57
2.5.1 线性链表的基本概念	57
2.5.2 线性链表的基本运算	63
2.5.3 循环链表	66
2.5.4 多项式的表示及其运算	68

2.6 数组与字符串	74
2.6.1 数组的顺序存储结构	74
2.6.2 规则矩阵的压缩	76
2.6.3 一般稀疏矩阵的表示	79
2.6.4 字符串	84
2.7 树与二叉树	89
2.7.1 树的基本概念	89
2.7.2 二叉树及其基本性质	92
2.7.3 二叉树的存储结构	95
2.7.4 二叉树的遍历	97
2.7.5 穿线二叉树	99
2.7.6 表达式的线性化	102
2.8 图	105
2.8.1 图的基本概念	105
2.8.2 图的存储结构	106
2.8.3 图的遍历	111
2.9 索引存储结构	114
2.9.1 索引存储的概念	114
2.9.2 “顺序-索引-顺序”存储方式	116
2.9.3 “顺序-索引-链接”存储方式	117
2.9.4 多重索引存储结构	118
习题	119
第3章 查找与排序技术	121
3.1 线性表的查找技术	121
3.1.1 顺序查找	121
3.1.2 有序表的对分查找	122
3.2 Hash 表技术	123
3.2.1 Hash 表的基本概念	123
3.2.2 几种常用的 Hash 表	126
3.3 线性表的排序技术	131
3.3.1 互换排序	131
3.3.2 插入排序	134
3.3.3 选择排序	137
3.3.4 其他排序方法简介	140
3.4 索引查找	143
3.4.1 分块查找	143
3.4.2 二叉排序树查找	145
3.4.3 B-树	149

3.4.4 B ⁺ 树	156
3.5 拓扑分类	158
习题	160
第4章 数据管理技术	162
4.1 数据管理技术的发展	162
4.1.1 数据管理技术发展的三个阶段	162
4.1.2 数据库管理系统	164
4.2 数据描述与数据模型	166
4.2.1 数据描述	166
4.2.2 数据模型	169
4.3 数据库体系结构	172
4.4 关系代数	174
4.4.1 关系代数运算	174
4.4.2 关系模型的查询优化	181
4.4.3 关系代数的等价变换	184
4.5 关系模式的规范化	185
4.6 数据库设计	189
4.6.1 数据库设计的过程	189
4.6.2 数据字典	196
4.6.3 数据库保护	198
4.7 关系数据库语言	199
4.7.1 关系数据库语言概述	199
4.7.2 数据库定义	202
4.7.3 数据查询	208
4.7.4 数据修改	215
4.7.5 数据控制	218
4.7.6 嵌入式 SQL	220
习题	224
第5章 Windows 程序设计	227
5.1 Windows 应用程序设计的特点	227
5.2 使用 Visual C++ 6.0	235
5.2.1 Visual C++ 6.0 可视化集成开发环境	236
5.2.2 创建工程和工作区	243
5.2.3 Win32 开发	246
5.2.4 MFC 编程	249
5.3 窗口、菜单与消息框	255
5.3.1 创建一个 Windows 程序	255

5.3.2	AppWizard 所创建的文件	260
5.3.3	编译和链接 Hi 程序	261
5.3.4	应用程序执行机制	262
5.3.5	几种窗口类型	266
5.3.6	菜单的使用	270
5.3.7	更新命令用户接口消息	276
5.3.8	快捷菜单	279
5.4	工具条与状态栏	281
5.4.1	工具条的可视化设计	281
5.4.2	工具条的编程技术	287
5.4.3	状态栏的设计与实现	291
习题		293
第 6 章	编译技术	294
6.1	编译程序的工作过程	294
6.2	状态矩阵法的编译过程	296
6.2.1	状态矩阵法的基本原理	296
6.2.2	状态矩阵的压缩	297
6.3	词法分析	299
6.3.1	词法分析的任务	299
6.3.2	读字符程序	301
6.3.3	状态矩阵法的词法分析过程	301
6.3.4	算术常数的识别和翻译	305
6.4	中间语言表示	308
6.4.1	波兰表示	308
6.4.2	三元组表示	310
6.5	语法的分析与加工	315
6.5.1	优先矩阵法	315
6.5.2	优先数法	317
6.5.3	状态矩阵法	318
6.5.4	递归子程序法	318
习题		320
第 7 章	应用软件设计与开发技术	322
7.1	软件工程概述	322
7.1.1	软件工程的观念	322
7.1.2	软件生命周期	322
7.1.3	应用软件开发的原则和方法	325
7.2	结构化分析方法	327

7.2.1 SA 方法的特点	327
7.2.2 数据流程图	329
7.2.3 数据字典	331
7.2.4 功能说明	332
7.3 结构化设计方法	334
7.3.1 SD 方法的特点	334
7.3.2 结构图	334
7.3.3 模块独立性的评价	336
7.4 测试与调试基本技术	341
7.4.1 测试	341
7.4.2 调试	349
7.5 软件开发新技术	351
7.5.1 原型方法	351
7.5.2 瀑布模型	352
7.5.3 面向对象技术	353
习题	355
参考文献	356

第1章 基础知识

1.1 计算机软件概述

1.1.1 计算机软件及其分类

计算机系统由计算机硬件系统和计算机软件系统组成。

计算机硬件系统是指实际的物理设备，包括计算机的主机和外围设备。

计算机软件系统，是指能指挥计算机工作的程序、程序运行时所需要的数据以及与这些程序和数据有关的文字说明和图表资料。其中文字说明和图表资料又称为文档。

相对于计算机硬件而言，软件是计算机的无形部分，但它的作用是很大的。这好比是人们为了看录像，就必须要有录像机，这是硬件条件；但仅有录像机还看不成录像，还必须要有录像带，这是软件条件。由此可知，如果只有好的硬件，但没有好的软件，计算机是不可能显示出它的优越性的。计算机软件技术实际体现了计算机应用和发展的水平。因此，软件是计算机系统的重要组成部分。

软件系统一般可以分为系统软件和应用软件两大类。

1. 系统软件

系统软件是指管理、监控和维护计算机资源(包括硬件和软件)，并提供用户与计算机之间界面等工具的软件。微型机中最常见的系统软件有操作系统、各种语言处理程序以及各种工具软件等。

(1) 操作系统

操作系统是最底层的系统软件，它是对硬件系统功能的首次扩充，也是其他系统软件和应用软件能够在计算机上运行的基础。

(2) 程序设计语言与语言处理程序

人们要利用计算机解决实际问题，一般首先要编制程序。程序设计语言就是用户用来编写程序的语言，它是人与计算机之间交换信息的工具。

程序设计语言是软件系统的重要组成部分，而相应的各种语言处理程序属于系统软件。程序设计语言一般分为机器语言、汇编语言和高级语言三类。

(3) 工具软件

工具软件又称为服务软件，它是开发和研制各种软件的工具。常见的工具软件有诊断程序、调试程序和编辑程序等。这些工具软件为用户编制计算机程序及使用计算机提供了方便。

① 诊断程序。诊断程序有时也称为查错程序，它的功能是诊断计算机各部件能否正常工作，因此，它是面向计算机维护的一种软件。例如，对微型机加电后，一般都首先运行 ROM 中的一段自检程序，以检查计算机系统是否能正常工作。这段自检程序就是一种最简单的诊

断程序。

② 调试程序。调试程序用于对程序进行调试，它是程序开发者的工具，特别是对于调试大型程序显得更为重要。例如，DEBUG 就是一般 PC 机系统中常用的一种调试程序。

③ 编辑程序。编辑程序是计算机系统中不可缺少的一种工具软件。它主要用于输入、修改、编辑程序或数据。

系统软件是计算机系统的必备软件。用户在购置计算机时，一般都要根据需要以及可能，配备相应的系统软件。

2. 应用软件

应用软件是指除了系统软件以外的所有软件。它是用户利用计算机及其提供的系统软件为解决各种实际问题而编制的计算机程序。由于计算机已渗透到了各个领域，因此，应用软件是多种多样的。

应用软件主要是为用户提供在各个具体领域中的辅助功能，它也是绝大多数用户学习、使用计算机时最感兴趣的内容。

应用软件具有很强的实用性，专门用于解决某个应用领域中的具体问题，因此，它又具有很强的专用性。由于计算机应用的日益普及，各行各业、各个领域的应用软件越来越多。也正是这些应用软件的不断开发和推广，更显示出计算机无比强大的威力和无限广阔的前景。

应用软件的内容很广泛，涉及到社会的许多领域，很难概括齐全，也很难确切地进行分类。

常见的应用软件有以下几种：

- ① 各种信息管理软件。
- ② 办公自动化系统。
- ③ 各种文字处理软件。
- ④ 各种辅助设计软件以及辅助教学软件。
- ⑤ 各种软件包，如数值计算程序库、图形软件包等。

1.1.2 程序设计语言及其语言处理程序

程序设计语言一般分为机器语言、汇编语言和高级语言三类。

1. 机器语言

机器语言是最底层的计算机语言。用机器语言编写的程序，计算机硬件可以直接识别。在用机器语言编写的程序中，每一条机器指令都是二进制形式的指令代码。在指令代码中一般包括操作码和地址码，其中操作码告诉计算机作何种操作，地址码则指出被操作的对象。对于不同的计算机硬件(主要是微处理器)，其机器语言是不同的，因此，针对一种计算机所编写的机器语言程序不能在另一种计算机上运行。由于机器语言程序是直接针对计算机硬件的，因此它的执行效率比较高，能充分发挥计算机的速度性能。但是，用机器语言编写程序的难度比较大，容易出错，而且程序的直观性比较差，也不容易移植。

2. 汇编语言

为了便于理解与记忆，人们采用能帮助记忆的英文缩写符号(称为指令助记符)来代替机

器语言指令代码中的操作码，用地址符号来代替地址码。用指令助记符及地址符号书写的指令称为汇编指令(也称符号指令)，而用汇编指令编写的程序称为汇编语言源程序。汇编语言又称符号语言。

汇编语言与机器语言一般是一一对应的，因此，汇编语言也是与具体使用的计算机有关的。由于汇编语言采用了助记符，因此，它比机器语言直观，容易理解和记忆，用汇编语言编写的程序也比机器语言程序易读、易检查、易修改。但是，计算机不能直接识别用汇编语言编写的程序，必须由一种专门的翻译程序将汇编语言源程序翻译成机器语言程序后，计算机才能识别并执行。这种翻译的过程称为“汇编”，负责翻译的程序称为汇编程序。

3. 高级语言

机器语言和汇编语言都是面向机器的语言，一般称为低级语言。低级语言对机器的依赖性太大，用它们开发的程序通用性很差，普通的计算机用户也很难胜任这一工作。

随着计算机技术的发展以及计算机应用领域的不断扩大，计算机用户的队伍也在不断壮大。为了使广大的计算机用户也能胜任程序的开发工作，从20世纪50年代中期开始逐步发展了面向问题的程序设计语言，称为高级语言。高级语言与具体的计算机硬件无关，其表达方式接近于被描述的问题，易为人们接受和掌握。用高级语言编写程序要比低级语言容易得多，并大大简化了程序的编制和调试，使编程效率得到大幅度的提高。高级语言的显著特点是独立于具体的计算机硬件，通用性和可移植性好。

目前，计算机高级语言已有上百种之多，得到广泛应用的有十几种，并且，几乎每一种高级语言都有其最适用的领域。表1.1列出了几种最常用的高级语言及其最适用的领域。

表 1.1

常用的高级语言

语言名称	适用范围
BASIC	教学和小型应用程序的开发
FORTRAN	科学及工程计算程序的开发
PASCAL	专业教学 and 应用程序的开发
C	中、小型系统程序的开发
COBOL	商业与管理应用程序的开发
dBASE	数据库管理程序的开发
FoxBASE	数据库管理程序的开发
C++	面向对象程序的开发
LISP	人工智能程序的开发
PROLOG	人工智能程序的开发
JAVA	面向对象程序的开发

必须指出，用任何一种高级语言编写的程序(称为源程序)都要通过编译程序翻译成机器语言程序(称为目标程序)后计算机才能执行，或者通过解释程序边解释边执行。

1.2 操作系统的基本概念

1.2.1 操作系统的功能与任务

操作系统是最基本的和核心的系统软件，也是当今计算机系统中不可缺少的组成部分，所有其他的软件都依赖于操作系统的支持。

从计算机系统的组成层次出发，操作系统是直接和硬件层相邻的第一层软件，它对硬件进行首次扩充，是其他软件运行的基础。操作系统实际上是由一些程序模块组成的，它们是系统软件中最基本的部分，其主要作用有以下几个方面：

① 管理系统资源，包括对 CPU、内存储器、输入输出设备、数据文件和其他软件资源的管理。

② 为用户提供资源共享的条件和环境，并对资源的使用进行合理调度。

③ 提供输入/输出的方便环境，简化用户的输入/输出工作，提供良好的用户界面。

④ 规定用户的接口，发现、处理或报告计算机操作过程中所发生的各种错误。

从上述几方面的作用可以看出，操作系统既是计算机系统资源的控制和管理者，又是用户和计算机系统之间的接口，当然它本身也是计算机系统的一部分。因此，概略地说，操作系统是用以控制和管理系统资源、方便用户使用计算机的程序的集合。

如果把操作系统看成是计算机系统资源的管理者，则操作系统的功能和任务主要有以下五个方面。

1. 处理机管理

处理机（即 CPU）是整个计算机硬件的核心。处理机管理的主要任务是：充分发挥处理机的作用，提高它的使用效率。

2. 存储器管理

计算机的内存储器是计算机硬件系统中的重要资源，它的容量总是有限的。存储器管理的主要任务是：对有限的内存储器进行合理的分配，以满足多个用户程序运行的需要。

3. 设备管理

通常，用户在使用计算机时或多或少地用到输入/输出操作，而这些操作都要涉及到各种外部设备。设备管理的主要任务是：有效地管理各种外部设备，使这些设备充分发挥效率；并且还要给用户提简单而易于使用的接口，以便在用户不了解设备性能的情况下，也能很方便地使用它们。

4. 文件管理

由于内存储器是有限的，因此，大部分的用户程序和数据，甚至是操作系统本身的部分以及其他系统程序的大部分，都要存放在外存储器上。文件管理的主要任务是：实现惟一地标识计算机系统中的每一组信息，以便能够对它们进行合理地访问和控制；以及有条理地组织这些信息，使用户能够方便且安全地使用它们。

5. 作业管理

作业管理是操作系统与用户之间的接口软件。它的主要任务是：对所有的用户作业进行

分类,并且根据某种原则,源源不断地选取一些作业交给计算机去处理。

从操作系统的以上五项主要任务可以看出,操作系统实际上确实是计算机系统资源的管理者。对于实际的一些操作系统,可能由于其性能、使用方式各不相同,使系统功能、基本结构、支持硬件和应用环境等方面也有所不同,因此,操作系统所要完成的任务也各不相同。例如,对于微机操作系统,作业管理就不需要,而重点是处理机管理、文件管理和设备管理。

1.2.2 操作系统的发展过程

操作系统是在计算机技术发展的过程中形成和发展起来的,它以充分发挥处理机的处理能力、提高计算机资源的利用率及方便用户使用计算机为主要目标。本小节简要讨论操作系统的形成和发展的过程,以及在每一发展阶段中系统的主要特点与采用的主要技术。

1. 手工操作阶段

在计算机发展的初期,计算机运行的速度比较低,能使用的外部设备也比较少,因此,在这一阶段中,用户以使用机器语言或符号语言的手编程序为主,操作员(也即用户)通过控制台上的开关来控制计算机的运行,而计算机通过指示灯来显示其运行的状态。在这种手工操作的方式下,由于大量的人工干预的存在,降低了计算机的使用效率,主要体现在以下几个方面:

① 由于单个用户独占计算机的所有资源,而造成资源得不到充分利用。

② 由于用户直接使用计算机硬件资源,因此,要求用户熟悉计算机各部分的细节,这就导致使用很不方便,也容易出错。

③ 由于进行手工联机操作,人工干预多,造成辅助时间长。

以上这些缺点在计算机发展的初期还不十分突出,但随着计算机运行速度的提高和计算机应用的日益广泛,与程序的实际运行时间相比,人工干预的时间所占的比例越来越大,以至于上机解题的时间大部分花在了手工操作上,并且,非计算机专业人员使用计算机的不方便性也越来越突出,因此,人工操作方式的弊病也就越来越明显了。

2. 成批处理系统

手工操作存在的根本问题是人工干预过多,因此,要克服手工操作方式的缺点,就必须减少人工干预,实现作业之间交接的自动化,以缩短作业交接时处理机的等待时间,从而较好地发挥计算机的效率。为此,就出现了成批处理系统。

在早期的成批处理系统中,操作员把若干个作业合为批,然后按先后次序通过输入机将这批作业读入内存缓冲区,再转储到磁带上,最后由监控程序再从磁带上顺序读入每个作业到内存,交给处理机去处理。在处理完一个作业之后,再读入下一个作业进行处理,直到全部输入并处理完这一批作业,再把下一批作业读入磁带,以同样的方式进行处理。由这个过程可以看出,在处理一批作业的时候,各个作业之间的交接是在监控程序的控制下自动完成的,因而缩短了由于手工操作所造成的处理机等待时间。在这种系统中,手工操作带来的矛盾得以缓和,但由于处理机和输入、输出设备是串行工作的,处理机与输入、输出设备之间速度不匹配的矛盾开始突出,处理机的绝大部分时间是处于等待状态。因此,为了进一步提高效率,出现了脱机技术。

在采用脱机技术的系统中,专门设置一台功能较弱、价格较低的小型卫星机来承担输入、

输出设备的管理任务。该卫星机只与外部设备相连，不与主机直接连接，因而称之为脱机批处理系统，如图 1.1 所示。

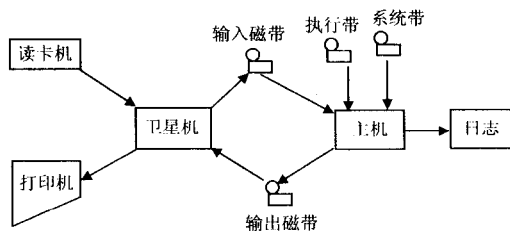


图 1.1 脱机批处理系统模型

在脱机批处理系统中，各个用户作业先由卫星机汇集到输入磁带上，然后由主机从磁带上把作业调到内存，并予以执行。主机在处理输入磁带上的作业时，将产生的输出结果送到输出磁带上，然后由卫星机把处理结果输出到输出设备上，每个作业的结果要在一批作业处理完以后才能得到。这就是脱机的单道程序成批处理系统。

脱机成批处理系统的特点是卫星机与主机并行工作，使主机摆脱了慢速的输入、输出操作。但是，它并没有从根本上解决发挥主机效率的问题，因为主机的运行和磁带机的输入、输出操作都是在程序控制下串行进行的，而磁带机的工作速度远低于主机，因此，主机的大部分时间仍消耗在输入/输出操作上。为了进一步提高系统的效率，有效的方法是使主机和输入、输出设备并行工作。为了使主机和输入/输出设备并行工作，后来人们使用了以通道技术、中断技术和多道程序设计技术为基础的假脱机（SPOOL）技术。

由此可以看出，成批处理系统较之手工操作具有很多的优点。这样的系统提高了计算机系统资源的利用率，特别是处理机的处理能力。同时，由于成批处理系统的出现，推动了软件系统的发展，并且为用户使用计算机提供了方便。

3. 执行程序系统

前面提到，为了充分发挥计算机的效率，必须处理好主机与外部设备在速度上不匹配的问题，因为慢速的输入/输出设备和主机串行工作，势必使主机经常处于等待状态，影响主机效率的发挥。为了解决这个问题，在硬件上引进通道和中断机构，这样就产生了主机和通道之间并行工作的执行程序系统。

通道是一种硬件机构，它独立于处理机而直接控制输入、输出设备与内存之间的数据传送。中断是外界（如输入、输出设备，通道等）向主机报告信息的一种通信方式。为了便于理解执行程序系统的工作原理，下面以在中断方式下进行输入、输出的执行过程为例，来说明执行程序系统的工作过程。

当主机需要在外部设备与内存之间传送数据时，首先在内存中开辟一个用于输入、输出的缓冲区，然后执行“启动通道”指令，此时就开始传送成组数据；在传送数据过程中，主机并不等待传送数据的完成，而是继续执行后续指令；当数据全部传送完成后，输入或输出设备报告给主机一个信号（称为中断请求信号），此时，主机暂时中断原来程序的执行，对其进行适当的处理（即执行中断处理程序），处理完后（即中断处理程序执行完）再继续执行被中断的程序。这个工作过程如图 1.2 所示。

在这种工作方式下，输入/输出设备与主机并行工作，但要求主机具有较强的中断处理能

力。所谓执行程序，包括输入/输出控制程序和中断处理程序。

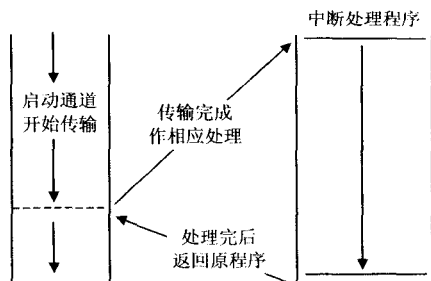


图 1.2 输入/输出设备与主机并行工作示意图

执行程序系统更充分地利用了系统资源，提高了计算机的执行效率，同时也增加了系统的安全性。另外，执行程序系统也简化了用户的使用界面，在程序设计时不必为时间匹配问题而花费精力。

4. 多道程序系统的引入

在执行程序系统阶段，各个用户程序在系统中是顺序执行的，因此，整个系统资源还不能得到充分利用。一般来说，用户程序往往各有处理的侧重点。例如，当执行计算量比较大的程序时，输入、输出设备的利用率就不高；而当计算的题目较小时，内存空间就得不到充分利用等等。如果在计算机中同时有多个程序在执行，就有可能使计算机系统各种资源得到更充分地利用。

所谓多道程序技术，是指在计算机内存中同时存放多道相互独立的程序，它们在操作系统的控制下，共享系统的硬件和软件资源。

在单处理机的系统中，并发程序在微观上只能是交替地运行，单在宏观上可看成是并行的。例如，在图 1.3 中，处理机同时在处理两道程序 A 和 B，但由于只有一个处理机，因此，实际上同时只能运行一个程序。假设先运行程序 A，当运行到 XA 处，由于某种原因运行不下去了，此时，处理机不再等待，而是转向运行程序 B，当运行到 XB 处，又由于某种原因运行不下去了，处理机将回到程序 A 的 XA 处运行（此时程序 A 有可能能运行了），如此继续。从图 1.3 可以看出，在某一小段时间内，处理机只能运行一道程序，但从宏观上来看程序 A 和程序 B 是在同时运行的，这正是这种系统的重要特征。

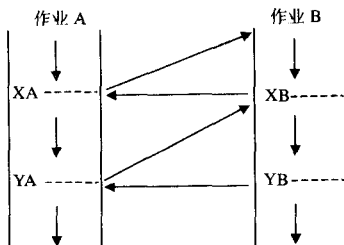


图 1.3 并发程序运行示意图

多道程序系统的显著优点是使主机、通道、输入/输出设备以及计算机系统的其他所有资源得到充分利用，也就是说，多道程序系统既具有并行性，又具有共享性。

1.2.3 操作系统的分类

计算机技术的迅速发展和日益广泛、深入的应用,必然会对操作系统的性能、使用方式等提出不同的要求,这就促使不同的操作系统在系统功能、基本结构、支援硬件和应用环境等方面都各有不同,从而形成不同类型的操作系统。

对操作系统进行分类的方法有很多。例如,按照计算机硬件规模的大小,可以分为大型机操作系统、小型机操作系统和微型机操作系统。它们的硬件资源和结构不同,使用环境也不一样,因此,操作系统设计的基本出发点、所要解决的问题和追求的主要目标都各不相同。如果按照操作系统在用户面前的使用环境以及访问方式,可以将操作系统分为多道批处理操作系统、分时操作系统和实时操作系统等。

1. 多道批处理操作系统

多道批处理操作系统包含“多道”和“批处理”两层意思。“多道”是指在计算机内存中存入多个用户作业。“批处理”是指这样一种操作方式,在外存中存入大量的后备作业,作业的运行完全由系统控制,用户与其作业之间没有交互作用,用户不能直接控制其作业的运行,通常称这种方式为批操作或脱机操作。

在多道批处理操作系统中,系统要根据一定的调度原则,从后备作业中选择一批搭配合理的作业调入内存运行。所谓“搭配合理”,主要是指要充分发挥系统资源的利用率,提高系统的处理能力,同时也要兼顾用户的响应时间。这类系统一般用于较大的计算机系统,由于它的硬件设备比较全,价格比较高,因此,多道成批处理系统十分注意 CPU 及其他设备的充分利用,以充分发挥资源的利用率为主要目标,追求高的吞吐量。这种系统的特点是,对资源的分配策略和分配机构以及对作业和处理机的调度等功能均经过精心设计,各类资源管理功能既全又强。

2. 分时操作系统

多道批处理系统吞吐量大,资源利用率高,但对用户来说,经常会感到使用不方便。用户一旦把他的作业交给系统之后,就失去了对该作业运行的控制能力。然而,对程序运行中的每个细节往往不是事先可以预料得到的,特别是新开发的程序,难免有不少错误或不当之处需要加以修改,因此,在这种情况下,用户希望能干预作业的运行,但多道批处理系统不提供这种程序调试的方便。此外,用户从提交作业到获得计算结果,一般需要等待很长的时间,因此,对于运行小程序的用户来说,大部分时间花费在交付作业与取结果上,延缓了程序的开发进程。针对多道批处理系统的这些缺点,又出现了一种使用户和程序之间可以有交互作业的系统,这就是分时操作系统。

在分时操作系统中,多个用户分享使用同一台计算机,即在一台计算机上联接若干台终端,每个用户可以独占一台终端。分时是指若干个并发程序对 CPU 的分时,其中每个程序对 CPU 的时间分享单位称为时间片。例如,设时间片长度为 100ms,现有 10 个用户,则操作系统对每个用户的平均响应时间为 $10 \times 100\text{ms} = 1\text{s}$ 。也就是说,每个用户轮流使用 100ms 的时间片。

分时操作系统具有以下几方面的特点:

- ① 同时性。即若干远、近程终端上的用户,在各自的终端上同时使用一台计算机。
- ② 独立性。即同一台计算机上的用户在各自的终端上独立工作,互不干扰。

③ 及时性。即用户可以在很短的时间内得到计算机的响应。

④ 交互性。即分时系统提供了人机对话的条件，用户可以根据系统对自己请求的响应情况，继续向系统提出新的要求，便于程序的检查和调试。

由上可知，分时操作系统显著提高了程序开发与调试的效率，为程序设计与开发者提供了一个理想的开发环境。并且，在分时系统下，用户可以通过终端随时使用本地或远程的计算机，使用很方便。此外，各分时系统的用户共享计算机资源，不仅使系统资源得以充分利用，还可以使用户之间方便地互相交流程序、信息和计算结果等，有利于用户之间合作完成一项计划。

第一个分时操作系统就是大家所熟悉的 UNIX 操作系统。

3. 实时操作系统

计算机的应用涉及各个领域和各个方面，其中信息处理和过程控制是计算机的重要应用领域，且都有一定的实时要求。这种具有实时要求的系统称之为实时系统。所谓实时，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。这里所说的外部事件是指计算机系统相连接的设备所提出的服务要求和数据采集。

实时系统分为实时过程控制系统和实时信息处理系统两类。前者用于工业生产的自动控制、导弹发射和飞机飞行等军事方面的自动控制、实验过程控制等。后者用于如机票预订管理、银行或商店的数据处理、情报资料查询处理等方面。这些实时系统的特点是严格的时间限制，它要求计算机对输入的信息作出快速响应，并在规定的时间内完成规定的操作。实时系统都要由适应这种要求的操作系统——实时操作系统进行管理和协调，以满足实际的需要。

4. 通用操作系统

根据实际需要，往往要将以上这些系统的功能组合起来使用，从而形成通用操作系统。例如，成批处理与分时处理相组合，分时作业为前台作业，而成批处理的作业为后台作业，这样，计算机在处理分时作业的空闲时间内，就可以适当处理一些成批作业，以避免时间的浪费，充分发挥计算机的处理能力。同样，成批处理系统也可以与实时系统相组合，此时，实时作业为前台作业，成批处理的作业为后台作业，这样也可以充分发挥系统资源的作用。

5. 优良的操作环境——多窗口系统

计算机特别是微型计算机已经普及到办公室和家庭，使用情况各不相同，使用者的水平也差别很大，因此，如何为用户提供一个最简单、最方便的操作环境，是推广和普及计算机应用的重要问题。要方便用户使用计算机，最重要的就是系统要向用户提供友好的界面，使用户能够通过简单、明了而且又非常醒目的提示，以尽可能少的键盘操作来使用计算机。多窗口系统正是这个目标的体现。

人们通过终端（或键盘与显示器）使用计算机，屏幕的输出管理是用户界面的重要部分。所谓多窗口系统最初基本上就是指管理屏幕上规定部分的输出和输入的工具。随着计算机技术的发展，多窗口系统的功能也在增强。实际上，现代的窗口操作系统已远远超出了上述的概念。所谓多窗口，就是把计算机的显示屏幕划分出多个区域，每个区域称为一个窗口，每个窗口负责处理和显示某一类信息。从不同的角度看，对多窗口系统可以有以下三种不同的认识：

① 从用户或应用的角度来看，多窗口系统是用户可以同时运行多道程序的一个集成化环境。