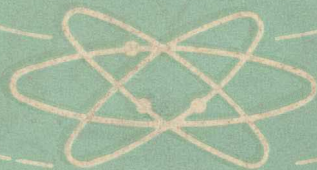


高等学校教材

# 计算机操作系统

汤子羸 杨成忠 等编



西北电讯工程学院出版社

高等学校教材

# 计算机操作系统

汤子瀛 杨成忠等 编著

西北电讯工程学院出版社

1986

## 内 容 简 介

本书专门介绍计算机系统的一个重要系统软件——操作系统。全书分为两部分。第一部分介绍操作系统原理，共分六章：第一章概括地讲述了操作系统的基本内容和概念；第二章系统地阐述了进程的概念，通信和调度方法；第三、四、五和六章介绍操作系统中各种资源的管理。第二部分介绍一个实例——UNIX操作系统，以第七、八、九和十章分别介绍UNIX系统中的存储、设备、文件和进程的管理。

本书是计算机专业的教科书，也可作为从事计算机工作的科技人员学习操作系统的参考书。

高等学校教材

### 计算机操作系统

汤子羸·杨成忠等 编著

---

西北电讯工程学院出版社出版

西北电讯工程学院印刷厂印刷

陕西省新华书店发行 各地新华书店经售

开本 787×1092 1/16 印张 20 8/16 字数 499 千字

1984年11月新一版 1986年11月第三次印刷 印数 50,001-70,000

---

统一书号：15322·54

定价：3.45元

## 出版说明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校工科电子类专业课教材的编审、出版的组织工作。从一九七七年底到一九八二年初，由于各有关院校，特别是参与编审工作的广大教师的努力和有关出版社的紧密配合，共编审出版了教材 159 种。

为了使工科电子类专业教材能更好地适应社会主义现代化建设培养人才的需要，反映国内外电子科学技术水平，达到“打好基础、精选内容、逐步更新、利于教学”的要求，在总结第一轮教材编审出版工作经验的基础上，电子工业部于一九八二年先后成立了高等学校《无线电技术与信息系统》、《电磁场与微波技术》、《电子材料与固体器件》、《电子物理与器件》、《电子机械》、《计算机与自动控制》，中等专业学校《电子类专业》、《电子机械类专业》共八个教材编审委员会，作为教材工作方面的一个经常性的业务指导机构，并制定了一九八二~一九八五年教材编审出版规划，列入规划的教材、教学参考书、实验指导书等 217 种选题。在努力提高教材质量，适当增加教材品种的思想指导下，这一批教材的编审工作由编审委员会直接组织进行。

这一批教材的书稿，主要是从通过教学实践、师生反映较好的讲义中评选择优和从第一轮较好的教材中修编产生出来的。广大编审者，各编审委员会和有关出版社都为保证和提高教材质量作出了努力。

这一批教材，分别由电子工业出版社、国防工业出版社、上海科学技术出版社、西北电讯工程学院出版社、湖南科学技术出版社、江苏科学技术出版社、黑龙江科学技术出版社和天津科学技术出版社承担出版工作。

限于水平和经验，这一批教材的编审出版工作肯定还会有许多缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评建议，共同为提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

## 前 言

本教材系由计算机与自动控制专业教材编审委员会计算机专业教材编审小组评选审定，并推荐出版。

该教材由杭州电子工业学院汤子瀛、哲凤屏，成都电讯工程学院杨成忠编著。重庆大学童颖教授担任主审。编审者均依据计算机编审小组审定的编写大纲进行编写和审阅的。

自1981年2月《操作系统原理》出版以来，经三届学生使用，积累了一些教学经验，也发现了教材中存在的一些问题。况且，该书是1979年着手编写的，五年以来，国际国内在操作系统领域中又取得了长足的进步，UNIX系统已是世界上最著名的分时系统之一，已成为PDP-11，VAX-11和Interdata 8/32等小型机系列以及M68000、Z8000、Inter8086等16位微型计算机的主要操作系统，国内外不少专家认为UNIX系统将成为16位计算机的标准操作系统。鉴于上述两个原因，我们借修编此书的机会，对原教材作了较大修改。再版操作系统教材，仍以操作系统原理为主，把RTOS实例换为UNIX系统，以使学生对目前世界上最有影响的操作系统有较深入的了解，以适应四化建设的需要。原书名改为《计算机操作系统》。

本书共分十章。原理部分共六章，基本内容与原版书一~六章相同。实例部分分为四章，即第七~十章。第七章较深入地阐述了存储器与处理机管理；第八章对块设备和字符设备的管理作了简要的介绍；第九章介绍UNIX系统中最有特色的文件系统；第十章是有关高级进程管理和控制等方面的内容。

本书是按100学时编写的，若按80学时讲授，建议免讲部分内容。

本书第一、二、五章由汤子瀛修改，第三、四、六章由杨成忠修改。实例部分的第八、十两章由汤子瀛编写，第七、九章由哲凤屏编写。这次参加本书审稿的有上海交通大学尤晋元同志，国防科技大学的李玉玺同志，重庆大学的梁光春、李朴、彭玉禄和苟荣秀等同志。清华大学的金兰教授、上海交通大学的张吉锋同志对本书的出版给予了很大的支持。

原《操作系统原理》教材出版后，得到了从事计算机科学和工程工作的同志们，特别是广大教师和学生的大力支持，提供了很多非常宝贵的意见，特在此表示衷心感谢。限于编者的水平，这次再版教材定然还存在着不少错误和不妥之处，仍恳请读者批评、指正。

编者 1984年3月



# 目 录

## 第一章 操作系统引论

1.1 单道批量处理系统	1
1.1.1 什么是作业	1
1.1.2 批量处理系统的引入	2
1.1.3 单道批量处理监督程序的功能和组成	3
1.2 多道批量处理系统	5
1.2.1 多道程序的引入	5
1.2.2 多道批量处理系统中管理程序的功能	6
1.2.3 虚拟处理机的概念	7
1.3 分时系统	8
1.3.1 引言	8
1.3.2 实现分时的方法	9
1.3.3 响应时间的改善	11
1.4 实时系统	13
1.4.1 实时系统的引入	13
1.4.2 实时系统的功能	13
1.5 计算机网络	14
1.5.1 计算机网络的发展过程	14
1.5.2 通信子网的功能	16
1.5.3 资源子网的功能	16
1.5.4 网络环境下的操作系统	17
1.6 操作系统的层次结构	18
1.6.1 操作系统的模块结构	18
1.6.2 有序分层法的概念和特点	18
1.6.3 层次设计的原则	19
1.6.4 层次间的调用方式	20
1.7 小结	21
习题	22

## 第二章 进程及处理机管理

2.1 进程的概念和定义	23
2.1.1 程序的顺序执行	23
2.1.2 程序的并行执行和资源共享	23
2.1.3 程序并行的特性	25
2.1.4 进程的定义和特征	26
2.1.5 小结	27

2.2 进程的状态和控制块	27
2.2.1 进程状态的变化	27
2.2.2 进程控制块 PCB	29
2.3 进程控制	30
2.3.1 创建原语	31
2.3.2 挂起原语	31
2.3.3 激活原语	32
2.3.4 撤消原语	33
2.3.5 阻塞原语和唤醒原语	33
2.4 进程的互斥与同步	34
2.4.1 临界区	35
2.4.2 测试与设置	36
2.4.3 信号量	38
2.4.4 进程同步	40
2.4.5 小结	43
2.5 进程通讯	43
2.5.1 消息缓冲	44
2.5.2 信箱通讯方式	45
2.6 进程调度	47
2.6.1 进程调度的功能和方式	47
2.6.2 进程调度与进程运行方式	48
2.6.3 最高优先级优先	49
2.6.4 轮转法	51
2.7 死锁	53
2.7.1 产生死锁的原因和必要条件	53
2.7.2 系统状态图和进程——资源图	55
2.7.3 死锁的检测	58
2.7.4 死锁的解除	60
2.7.5 死锁的预防	62
习题	64

## 第三章 作业管理

3.1 引言	66
3.2 用户与操作系统之间的接口	67
3.3 系统调用命令	67
3.3.1 什么是系统调用	67
3.3.2 系统调用执行过程	69
3.3.3 系统调用命令简介	69

3.4 作业的组织 .....	71	4.6 请求页式存储管理 .....	133
3.4.1 作业、作业步、作业流 .....	71	4.6.1 实现原理 .....	134
3.4.2 作业类别 .....	72	4.6.2 外页面表 .....	135
3.4.3 批量型作业的组织 .....	72	4.6.3 置换算法 .....	136
3.5 作业控制 .....	73	4.6.4 页面共享 .....	138
3.5.1 脱机作业控制 .....	73	4.6.5 请求页式系统的优缺点 .....	139
3.5.2 联机作业控制 .....	77	4.6.6 分页系统小结 .....	140
3.6 作业的管理和调度 .....	78	4.7 分段存储管理 .....	140
3.6.1 作业的状态和处理流程 .....	78	4.7.1 分段地址空间 .....	140
3.6.2 作业的输出与作业的输出 .....	81	4.7.2 实现原理 .....	141
3.6.3 作业调度的功能与算法的评价 .....	83	4.7.3 保护措施 .....	143
3.6.4 单道批处理系统的作业调度 .....	84	4.7.4 段式虚拟存储系统 .....	143
3.6.5 多道程序环境中的作业调度 .....	86	4.7.5 分段的动态链接 .....	145
3.7 小结 .....	92	4.7.6 分段的共享 .....	149
附录 多道程序环境下处理机等待百		4.7.7 分段管理的优缺点 .....	150
分比的计算 .....	96	4.8 段页式存储管理 .....	151
习题 .....	93	4.8.1 实现原理 .....	151
<b>第四章 存储管理</b>		4.8.2 管理算法 .....	153
4.1 引言 .....	98	4.8.3 段页式系统的优缺点 .....	154
4.1.1 存储管理的目的和功能 .....	98	4.8.4 段式系统小结 .....	154
4.1.2 存储分配 .....	99	习题 .....	155
4.1.3 重定位 .....	100	<b>第五章 设备管理</b>	
4.1.4 虚拟存储器概念的引入 .....	104	5.1 引言 .....	156
4.1.5 小结 .....	105	5.1.1 I/O 设备的类型 .....	156
4.2 单一连续区分配 .....	105	5.1.2 设备管理的任务和功能 .....	156
4.3 分区式分配 .....	107	5.2 通道技术 .....	157
4.3.1 固定式分区 .....	107	5.2.1 I/O 控制方式的演变 .....	157
4.3.2 可变式分区 .....	108	5.2.2 通道的类型 .....	159
4.3.3 可重定位分区分配 .....	117	5.2.3 并行操作 .....	160
4.3.4 多重分区分配 .....	119	5.2.4 “瓶颈”问题 .....	160
4.3.5 分区的保护措施 .....	120	5.2.5 通道指令和通道程序 .....	161
4.3.6 分区分配的优缺点 .....	121	5.3 缓冲技术 .....	162
4.3.7 小结 .....	122	5.3.1 缓冲的引入 .....	162
4.4 覆盖和交换 .....	122	5.3.2 单缓冲和双缓冲 .....	163
4.4.1 覆盖管理 .....	122	5.3.3 多缓冲 .....	163
4.4.2 交换技术 .....	126	5.3.4 缓冲池 .....	164
4.4.3 小结 .....	128	5.3.5 小结 .....	167
4.5 分页存储管理 .....	128	5.4 设备分配程序 .....	167
4.5.1 实现原理 .....	129	5.4.1 设备管理中的数据结构 .....	167
4.5.2 关于页表的一些考虑 .....	131	5.4.2 设备分配原则 .....	169
4.5.3 分页存储分配算法 .....	132	5.4.3 设备分配程序 .....	171
4.5.4 分页存储管理的优缺点 .....	133	5.5 I/O 设备的处理程序 .....	173

5.5.1 处理机与外部设备间的通讯方式	173
5.5.2 中断机构	173
5.5.3 设备处理方式	176
5.5.4 小结	178
5.6 I/O 控制系统	179
5.6.1 设备管理的数据结构	179
5.6.2 Request 过程	179
5.6.3 Attach 和 Detach 过程	180
5.6.4 Release 过程	181
5.6.5 Read 过程	182
5.6.6 I/O 进程	183
习题	186

## 第六章 文件管理

6.1 引言	187
6.2 文件、文件系统	187
6.2.1 文件	187
6.2.2 文件系统	189
6.3 存取方法和文件的逻辑组织	190
6.3.1 顺序存取方法	190
6.3.2 直接存取方法	192
6.3.3 按键存取	192
6.4 文件的物理组织	193
6.4.1 文件的物理结构	193
6.4.2 文件类型与文件存储设备、存取方法的关系	197
6.4.3 多级索引	198
6.4.4 逻辑记录与物理块大小不等带来的影响	199
6.5 文件存储器存储空间的管理	202
6.5.1 空白文件目录	202
6.5.2 空白块链	203
6.5.3 位示图	205
6.6 文件目录	206
6.6.1 单级文件目录	206
6.6.2 二级目录	208
6.6.3 多级目录	209
6.6.4 便于共享的目录组织	210
6.6.5 符号文件目录的查寻技术	211
6.6.6 文件目录的管理	215
6.7 文件的存取控制	215
6.7.1 存取控制矩阵	216

6.7.2 存取控制表	217
6.7.3 用户权限表	217
6.7.4 口令	218
6.7.5 密码	218
6.7.6 文件保护机构的例子	218
6.8 文件系统的一般模型	220
6.8.1 用户接口及初始化模块	220
6.8.2 符号文件系统 (SFS)	221
6.8.3 基本文件系统 (BFS)	221
6.8.4 存取控制验证 (ACV)	222
6.8.5 逻辑文件系统 (LFS)	222
6.8.6 物理文件系统 (PFS)	222
6.8.7 分配策略模块 (ASM)	223
6.8.8 设备策略模块 (DSM)	223
6.8.9 I/O 调度和控制系统	223
6.8.10 文件系统模块之间调用与返回	223
6.9 文件的使用	224
6.9.1 建立文件	225
6.9.2 打开文件	225
6.9.3 读文件	226
6.9.4 写文件	226
6.9.5 关闭文件	226
6.9.6 撤消文件	226
6.10 小结	227
习题	227

## 第七章 UNIX存储器与处理机管理

7.1 UNIX 系统概述	229
7.1.1 UNIX 系统的发展历史	229
7.1.2 UNIX 系统的主要特点	229
7.1.3 UNIX 系统的组成	230
7.1.4 UNIX 系统的功能	230
7.1.5 UNIX 系统的结构	231
7.2 存储管理的硬件支持	232
7.3 进程映象的组成和装配	233
7.3.1 进程映象的组成	233
7.3.2 核心程序的装配	235
7.3.3 进程映象在虚拟空间中的装配	235
7.3.4 进程映象在物理空间中的装配	236
7.4 地址映射	237
7.4.1 可变页式地址映射	237
7.4.2 装配样本映射寄存器	238
7.4.3 装配页面映射寄存器	239



7.5 存储器的分配与释放	229
7.5.1 存储器的分配	240
7.5.2 存储器的释放	240
7.6 进程调度	241
7.6.1 保护下退进程的运行环境	241
7.6.2 分配处理机	243
7.6.3 恢复上升进程的运行环境	244
7.7 进程通讯	244
7.7.1 sleep (chan, pri)过程	244
7.7.2 setrun (p)过程	245
7.7.3 wakeup (chan)过程	246
7.7.4 进程状态转换	246

## 第八章 UNIX设备管理

8.1 引言	247
8.1.1 块设备与字符设备	247
8.1.2 设备管理的特点	247
8.2 设备管理中的数据结构	248
8.2.1 块设备缓冲区和缓冲控制块	248
8.2.2 字符设备的缓冲池	250
8.2.3 设备控制表	250
8.2.4 设备开关表	251
8.3 缓冲池的管理	252
8.3.1 块设备缓冲池的管理	252
8.3.2 字符设备的缓冲池管理	253
8.4 块设备管理	255
8.4.1 RK磁盘寄存器	255
8.4.2 RK磁盘驱动程序	256
8.4.3 I/O 队列管理程序	257
8.4.4 磁盘读、写程序	257
8.4.5 磁盘中断处理程序	260
8.4.6 块设备管理小结	261
8.5 字符设备管理	262
8.5.1 纸带机设备寄存器	262
8.5.2 纸带机设备驱动程序	263
8.5.3 纸带机打开和关闭程序	363
8.5.4 纸带机读、写程序	264
8.5.5 纸带机中断处理程序	265
8.5.6 字符设备管理小结	265

## 第九章 UNIX文件系统

9.1 文件系统概述	267
9.1.1 UNIX文件系统的特点	267

9.1.2 文件分类	267
9.1.3 文件目录结构	268
9.1.4 文件系统的资源管理观点	269
9.2 索引结点的管理	269
9.2.1 卷资源表	269
9.2.2 索引结点	270
9.2.3 索引结点的分配与释放	272
9.3 空闲盘块的管理	275
9.3.1 空闲盘块的组织	275
9.3.2 盘块空间的寻址方式	275
9.3.3 地址映象	276
9.3.4 空闲盘块的分配	278
9.3.5 空闲盘块的释放	279
9.4 目录文件的管理	281
9.4.1 目录文件及有关的变量	281
9.4.2 构造目录	282
9.4.3 检索目录	283
9.5 文件表的管理	286
9.5.1 用户打开文件表	286
9.5.2 系统打开文件表	286
9.5.3 用户打开文件表和系统打开文件表的分配	287
9.5.4 打开与关闭文件的过程	287
9.6 文件的读、写管理	289
9.6.1 用户区和缓冲区之间的信息传送	289
9.6.2 文件的读、写过程	291
9.7 系统调用及其实现过程	293
9.7.1 系统调用	293
9.7.2 文件的创建和联结	294
9.7.3 文件的打开和关闭	296
9.7.4 文件的解除联结和删除	297
9.7.5 文件的读和写	298
9.8 小结	299

## 第十章 UNIX进程的管理与控制

10.1 进程映象的对换	301
10.1.1 swap 过程	301
10.1.2 xswap 过程	302
10.1.3 sched 过程	303
10.2 进程映象的管理	305
10.2.1 数据段的扩大和缩小	305

10.2.2	正文段的连接和释放 .....	306	10.4	软中断 .....	314
10.2.3	建立子进程映象 newproc过程 .....	308	10.4.1	软中断的概念及分类 .....	314
10.3	进程控制 .....	310	10.4.2	软中断处理方式的预置 .....	315
10.3.1	建立子进程 .....	310	10.4.3	软中断的发送 .....	316
10.3.2	执行一个文件 .....	311	10.4.4	软中断的检查与进入 .....	316
10.3.3	进程自我终止 .....	313	10.5	跟踪 .....	318
			10.6	小结 .....	320

# 第一章 操作系统引论

随着计算机的发展，计算机系统的硬件和软件资源也就愈来愈丰富。为提高这些资源的利用率和增强系统的处理能力，最初出现的是监督程序。它是用户与计算机之间的接口，即用户通过监督程序来使用计算机。到六十年代中期，监督程序又进一步发展，形成了操作系统。操作系统具有下述三方面的功能：(1)控制和管理计算机系统的硬件和软件资源，使之得到有效的利用；(2)合理地组织计算机系统的工作流程，以增强系统的处理能力；(3)提供用户与操作系统之间的软件接口，使用户能通过操作系统方便地使用计算机。因此，我们可以这样说：操作系统是控制和管理计算机硬件和软件资源，合理地组织计算机工作流程以及方便用户的程序的集合。

由于操作系统的主要功能是管理计算机系统资源，其中大部分程序都属于资源管理程序。因此，可以从操作系统是资源管理程序这样的观点出发，来研究操作系统（从而形成了所谓的资源管理观点）；又因为通常把系统中的资源成分四类，即处理机、存储器、外部设备和信息（程序和数据等），因而也把操作系统分成这样几个相应的部分，即(1)处理机管理模块；(2)存储器管理模块；(3)设备管理模块；(4)信息管理模块。目前绝大多数操作系统都包含有这四大模块。此外，为了合理地组织工作流程和方便用户，系统又提供了作业管理模块，用于对作业进行控制和管理。本书也主要是从资源管理的观点，把操作系统原理部分分为六章，第一章是操作系统引论，第三章介绍作业的组织、控制和调度，其余四章分别介绍各类资源的管理；本书的后四章介绍一个操作系统的实例——UNIX系统，基本上也是按资源管理观点来分章的。

本章是从用户观点对操作系统作一般介绍。通过对单道批量处理系统、多道批量处理系统、分时系统、实时系统和计算机网络的介绍，使读者了解计算机操作系统和网络环境下的操作系统的形成和发展过程；了解各种类型操作系统中所要解决的主要问题，以及解决这些问题的方法；也使读者对操作系统中的一些基本概念有一初步了解；由于现代操作系统大都采用层次结构，故我们又对操作系统的层次结构和有序分层法做了简单的介绍，以便以后再从层次观点来介绍操作系统。

## 1.1 单道批量处理系统

### 1.1.1 什么是作业

通常，一个用户程序是有一定独立性的程序段的有序集合。这些程序段都是以某种语言形式写出的，例如用ALGOL、FORTRAN或汇编语言等。在一个用户程序中各程序段可用不同的语言。然而机器在执行时只能识别机器语言。因此每个程序段在执行前，都必须首先被翻译成机器语言。为使机器知道所用的语言形式，又必须以某种命令形式（作业控制卡或作业说明书）告诉系统，以便系统调用相应的翻译程序将不同的语言翻译成机器语言。此外，当用户程序开始时，同样需要以某种命令形式告诉系统。用户程序及其所需的数据和命令一起形成一个作业(Job)。

逻辑上作业是由有序的作业步组成，每个作业步又是由完成作业中某一相对独立事件的程序和数据构成的，并由命令定义之。例如，一个作业可由“翻译 FORTRAN 源程序为机器语言”，“装入已编译的程序”，“执行该目标程序”三个命令所定义三个作业步组成。

在批量处理环境中，通常把一批作业有序地排在一起形成一个作业流，并采取命令形式来标识一个新作业的开始和前一作业的开始。

### 1.1.2 批量处理系统的引入

在五十年代，计算机硬件大都由电子管构成，其体积庞大，内存容量小且速度慢，每秒只能运行几千次。软件也处于低级发展阶段，仅有汇编语言以及少量的服务性程序，计算机主要用于科学计算。

此时计算机的工作基本上采用人工操作，操作员将纸带（或卡片）装进纸带输入机（或卡片输入机），把程序和数据输入，然后通过控制台开关启动程序。在程序运行中也是利用控制台开关对它进行控制的。仅当程序完成并取走它的纸带和计算结果后，才让下一个用户上机操作。这种人工操作方式具有如下特点：

**1. 用户独占全机** 一个计算机为一个用户独占，其全部资源由他一人支配。因此用户可以较方便地使用资源，不会出现因资源为别的用户占用而等待的现象，但资源利用率却非常低。因为很多小型题无需利用计算机所提供的全部资源。

**2. CPU等待人工操作** 用户所使用的纸带等，仅在上机时才装入相应的设备上，在此期间机器是空闲的。同样，当计算完成后，在做卸带取卡等人工操作时，机器也无事可做，因此机器的利用极不充分，特别是在运行短程序时显得更为突出。

以上两点充分说明人工操作方式严重地损害了资源的利用率，此即所谓的人机矛盾。对第一代计算机来说，人机矛盾尚不突出，因为计算机本身所拥有的资源并不多，内存容量一般只有几 k，I/O 设备也只有几台。此外，由于计算速度低，计算所需时间相对较长，因而人工操作时间所占比例也不算太大。但随着计算机规模的不断扩大，例如，内存容量由几 k 增至几十 k 以至几百 k，I/O 设备由几台增至几十台，计算机的速度由每秒几千次增至百万次，人机矛盾就变得严重起来。如人工操作方式再不改变，资源利用率可能降为百分之几，甚至更低。

此外，随着 CPU 速度的大幅度提高，使得 CPU 和 I/O 设备间速度不匹配的矛盾更为突出。为了解决速度不匹配的矛盾，并使它们之间能重叠操作，又出现了一个关键性硬件——通道设备。而缓冲技术特别是脱机输入输出操作，又进一步缓和了 CPU 和 I/O 设备间速度的矛盾。早期的脱机输入操作是用户程序和数据在一台外围计算机的控制下，预先从低速输入设备输入到磁带上，当 CPU 需要这些程序和数据时，就可直接从磁带调入内存。在脱机输出时，CPU 只要把结果送至磁带上，然后再在另一台外围机的控制下，使之由相应的输出设备输出，这样就大大加速了输入输出过程。

在脱机输入输出方式中，由于事先已把若干个作业记录在一盘磁带上，这意味着用户程序的处理顺序已经排定，机器先处理磁带上的第一个作业，即把它由磁带传送至内存并启动之，把控制权交给作业 1。当作业 1 完成后又把控制权还给系统，系统再把磁带上的下一个作业输入内存并启动之，把控制权交给第二个作业。按这种方式对磁带上的作业自动地一个接一个地进行处理，于是便形成了批量处理系统。不难看出，单道批量处理系统是在解决人

机矛盾和CPU-I/O速度矛盾的过程中,也就是在提高资源利用率的过程中发展起来的。此时,为对系统中的作业和计算机各种资源进行监督和管理,又相应地配置了监督程序(Monitor)或管理程序(Supervisor),它不仅是操作系统的前身,而且也是操作系统的核心部分。

### 1.1.3 单道批量处理监督程序的功能和组成

早期单道批量处理系统中的配置如图 1-1 所示。监督程序放在内存的顶部或底部。记录由编译程序、汇编程序、装入程序等组成的系统程序的磁带称为系统带。作业流由卡片机输入,源程序被翻译为目标程序后暂存在磁带上,这种带称为目标带。

该批量处理系统的主要目的是使整个作业流能自动地、顺序地运行,以节省人工操作时间和改善机器的利用情况。当作业都是短小任务时,批量处理能获得良好的效果。在最基本的批量处理形式中的工作流程如图 1-2 所示。

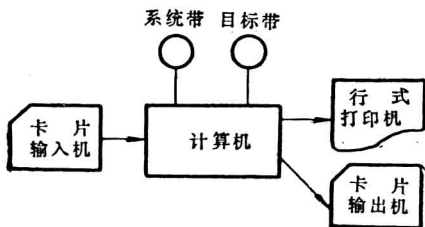


图 1-1 早期单道批量处理系统的组织

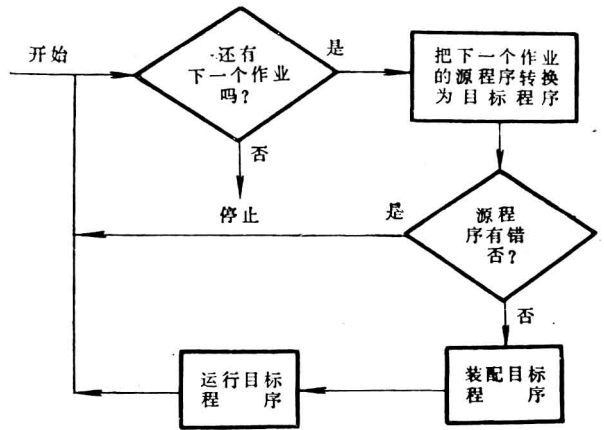


图 1-2 单道批量处理流程图

**1. 作业控制部分** 为使监督程序能代替操作员对作业进行控制和管理,用户必须通过某种命令方式,把与作业有关的信息提供给系统。例如,作业命令JOB,用以标识一个作业的开始;结束命令END,用以标识一个作业或作业步的结束;执行命令EXECUTE,表示用户希望执行自己的用户程序或调用系统程序。系统愈完善,能处理的事情愈多,需要的信息也愈多。当系统接到这些命令后,首先对该命令作正确性检查,若无误,再进一步通过查找一张命令表,而转至相应的命令处理程序,如编辑程序。在该命令表中包括了所有的命令,在每个命令表中记录有命令标识符及该命令程序的入口(在内存或外存中的地址)。这样,当系统接到命令后便到该表中去查找。若查到,便可转至相应的命令处理程序;若在该表中查不到发来的命令,则认为该命令非法。为实现这些功能,系统中必须配置相应的命令解释程序(Command Interpreter)。

在批量处理环境下,系统还必须能根据命令给出的信息,识别一个作业或作业步的开始和结束,以便当一个作业或作业步完成时,无须操作员的干预便可自动地过渡到下一个作业或作业步。这就需要有一个作业定序程序(Job Sequencer),与命令解释程序相配合来完成这一任务。在定序程序中利用了一张作业描述表来记录作业标识符、估计运行时间和有关作业的某些限制(最大运行时间,打印行数)以及作业的累计费用等。当一个作业完成时,定序程序便检索该表,从中找出下一个作业,把它从外存调入内存投入运行。由命令解释程序和定序程序两部分,组成了监督程序中的作业控制部分。

**2. 内存管理** 由于内存容量有限，因此常把一些系统程序和用户程序放在后援存储器中。例如，当系统接到一条编译源程序的命令时，便首先从后援存储器中把相应的编译程序调入内存，为此，还必须事先为编译程序找到一个内存空间来存放它。由于通常在内存中并没有足够的空闲空间容纳编译程序，因此必须将原来内存中的其它程序调至后援存储器，以获得足够大的空闲空间。为此系统必须知道被编译之源程序的所在位置，记住被编译为机器语言的目标程序所在地址，以备需要时查找。此外，当源程序被翻译成目标模块后，还需把它们装配成一个完整的用户目标程序。同样，系统也应首先找到内存空间来存放它，然后才能开始执行。由此可知，监督程序需要具备内存分配的功能。内存分配是借助于一张内存使用表来实现的，表中记录了哪些内存区域已经使用，哪些是空闲空间。当一个作业到达时可通过查找内存使用表，为该作业找到空闲的内存空间。

**3. 设备管理** 用户程序和它所需要的数据以及相应的控制命令，都需要通过输入机输入。当作业执行后，又要求把所得的结果以打印或穿孔的形式输出。为使操作员了解作业进展情况，每当作业和作业步在开始或结束时，都应打印出相应的信息通知操作员。此外，虽然在正常情况下无需人工干预，但在异常情况下仍需人工干预。这就需要把作业在执行中的有关信息、状态通知操作员。因此，读写控制是监督程序应具备的基本功能。

当一个用户希望把他所得到的结果作为永久性或暂时性记录保存在系统中时，系统便将该结果信息组成文件，存放在磁带或磁盘上。文件通常被认为是有关联的数据记录的集合，记录则是有序字符的组合。例如，文件可以是用户的源程序或某单位职工的名册。在这种情况下，甚至各种低速 I/O 设备，例如纸带输入机，打印机等，也都可看做是按字符流形式存取的文件。因此，无论是文件或 I/O 设备都可采取同一方式，用符号名称进行访问。文件管理的主要目的是为用户提供一种存储信息和检索信息的手段。为了记录每个外存的使用情况系统为每个外存准备了一张使用表。另外，系统中还保存了一张记录每个文件所在地址的文件目录。这样，当用户需要检索一个文件时，系统便可查阅文件目录，从而迅速地把文件调出。事实上，文件管理的方法极象图书馆中图书的管理。

在系统中具体的 I/O 操作由 I/O 设备管理程序中的第三部分——设备处理程序来实现。这些操作包括启动指定的 I/O 设备，处理来自 I/O 设备的中断等。为协助设备处理程序来完成具体 I/O 操作而设置了一张设备控制表 DCT，它记录了每个 I/O 设备的标识符、现行状态（忙闲）、连接的控制器等。

综上所述不难看出，无论是作业控制，内存管理，还是设备管理都是借助于一些表格来实现的，它们是监督程序实施管理功能的数据基础，故通常称之为数据基 (Data Base)。图 1-3 示出了监督程序的粗框图。

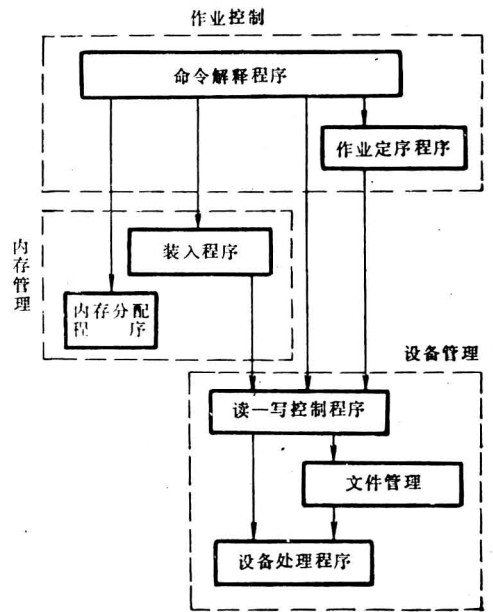


图 1-3 监督程序的粗框图



## 1.2 多道批量处理系统

### 1.2.1 多道程序的引入

**1. 提高处理机的利用率** 虽然单道批量处理系统已大大减少了人工操作的时间，提高了机器的利用率。在理想情况下，操作员所做的全部工作，是把所有作业适当地分批，装入相应的输入机，并启动之；随后，作业便一个接着一个地执行，最后把所得的结果取走。即使如此，对于某些作业在发出 I/O 请求后，还必须等待，而在单道程序运行中的作业，等待就意味着机器空闲，特别是由于 I/O 设备的低速性，将使机器的利用率变得非常低。图 1-4 示出了单道程序运行情况。在  $t_1$  时刻，用户程序发出 I/O 请求后进入监督程序，作适当处

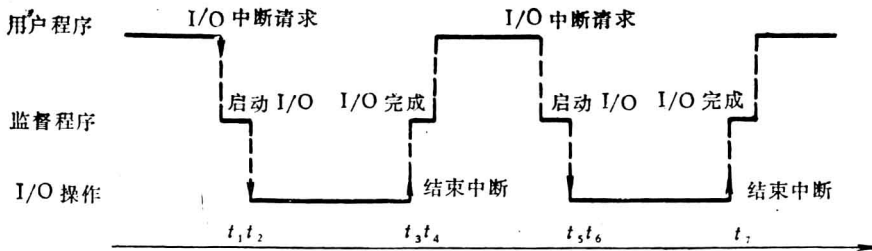


图 1-4 单道程序运行情况

理后在  $t_2$  时刻启动 I/O 设备工作，用户程序处于等待 I/O 操作完成的状态，在  $t_3$  时刻 I/O 完成，发出结束中断信号后进入监督程序， $t_4$  时刻又恢复用户程序运行（图中  $t_2 \sim t_3$ 、 $t_6 \sim t_7$  为 CPU 空闲时间）。

为了改善 CPU 的利用率引入了多道程序，即同时把几个作业放入内存，它们分时共用一台计算机。CPU 先对第一道程序进行处理，当它需要 I/O 时，CPU 处理了它的 I/O 请求后便转向第二道程序，使第一道程序的 I/O 操作和第二道程序的处理并行。当第二道程序需要 I/O 时，CPU 处理了它的 I/O 请求后又转向第三道程序，使第三道程序的处理与第一、二道程序的 I/O 操作并行。图 1-5 示出了两道程序的运行情况（其中  $t_4 \sim t_5$ 、 $t_8 \sim t_9$  为 CPU 空闲时间）。

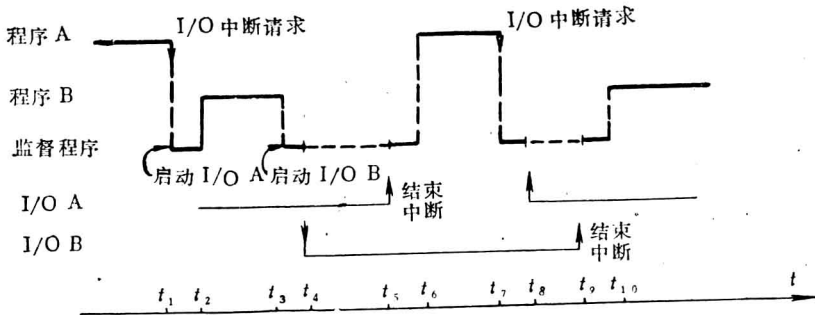


图 1-5 两道程序运行情况

**2. 改善内存和 I/O 设备的利用情况** 随着计算机的发展，具有各种用途的 I/O 设备也愈来愈多，在单道程序运行时，很难充分利用所有的 I/O 设备。此外，为适应日益膨胀的大型作业的需要，内存容量也在不断扩大。

但作业并非总是大型的，经常遇见的倒是中、小型作业，这样，就会有很大内存空间未能装入有用信息而白白浪费。

**3. 人工干预** 在单道批量处理系统中，运行情况往往不象上述那样理想，即作业一个接一个地往下运行而无需人工干预。实际上，用户程序经常出现这样或那样的错误，如“越界”、“地址错”、“阶上溢”等。一旦出现了这些错误，程序便无法运行下去，须经操作系统分析原因后，或从头做起(复执)，或暂停下来等待人工干预，这又意味着处理机时间的损失。但在多道程序环境下，当某程序出错时，系统把现场记录下来后便马上转向另一道程序而继续运行，不致严重影响系统效率。

**4. 人机会话** 在现代计算机中广泛采用人机会话。所谓人机会话，是指用户通过终端设备，如控制台打字机，向机器发布各种命令，系统通过用户终端设备向用户回答或提出问题，这无论对新程序的调整，还是对计算机的辅助设计都带来极大方便。但人机会话的速度很低，当人用控制台打字机向机器发布命令时，每秒钟只能发送几个字符，仅在一条命令送完后机器才予以响应。因此，在单道程序运行时机器效率受到严重影响。然而在多道程序环境下，命令的输入是穿插在其他程序中进行的。宏观地看，机器一面运行程序，一面做缓慢的人机会话，这就不至于严重地影响机器效率。

### 1.2.2 多道批量处理系统中管理程序的功能

由上所述可知，多道批量处理系统是利用 CPU 的等待时间来运行其他程序，这就显著地提高了资源的利用率，从而增强了系统的吞吐能力。因此，现代的大、中、小型计算机广泛地采用多道程序设计。但它也出现了一些新的问题，只有将这些问题妥善地解决好，才能有效地实现多道程序运行。其中的基本问题，是用户提交给系统的作业所需要的资源总和，远比系统所拥有的资源多，使得系统无法同时满足所有作业的资源要求而将它们开动起来。于是，这些作业必将争夺资源，以便投入运行。为使多道程序能有条不紊地运行，系统中必须增设管理程序，以便把这些资源管理起来，遵循一定的策略把这些资源分配给某些作业。按照系统资源的类型可把管理程序分为四种：

**1. 存储器管理** 在单道程序运行时，系统的全部资源为正在执行的作业所独占。然而在多道程序运行时，由于内存中同时装有若干道程序，故内存管理的第一个功能，是按一定策略进行内存分配，使它们不致因相互重叠而丢失信息。第二个功能，是必须防止因某道程序出现异常情况而破坏其它程序。当然，所有的用户程序更不能破坏管理程序，这就要求存储管理提供存储区保护的功能。

当一个作业所要求的内存容量或几个作业所要求的内存容量总和，超过实际内存容量时，为使这些作业都能投入运行，内存管理的第三个功能，是必须具有内存扩充的能力，亦即系统中应采用虚拟存储器技术。所谓虚拟存储器技术，是指通过某种机构，把由内存和后备存储器组成的两级存储器系统，变成可由用户程序直接存取的一级存储器系统。通常由于磁盘的容量远大于内存容量，因此用户所看到的内存容量就比物理内存大。这就使得一个大的用户程序可以在这样的系统中运行，该系统所具有的物理内存比实际需要小得多。

**2. 处理机管理** 在单道程序运行时，处理机只为一道程序所占用，而在多道程序运行时，处理机已为多道程序所共享，这样，必将引起各道程序对处理机的争夺，因此，处理机管理所要解决的中心问题是如何把处理机分配给进程。所谓进程，是指在系统中能独立运行

的基本程序单位，因而也是作为资源分配的基本单位。关于进程的概念将在第二章中详细介绍。

将处理机分配给进程的过程，通常可分为三个阶段：(1)提交-收容：用户通过输入机将作业提交给系统，由装入控制程序为它分配外存空间，并把它装入到指定地点。这时该作业便处于收容状态，并称之为后备作业；(2)收容-就绪：由作业调度程序按一定的调度算法，从后备作业中选取一批作业，为它们分配I/O设备和存储空间，调入内存并为之创建进程。此时这些进程已获得除处理机外的其它必要资源，我们称此时的进程处于就绪状态，将它们插入到就绪队列，使之成为具有获得处理机资格的候选进程；(3)就绪-运行：由进程调度程序按一定算法做进一步仲裁，即从就绪队列中选出一个进程，把处理机分配给它，使之投入运行。

**3. 设备管理** 在多道程序运行时，同样也会发生几道程序对外部设备的争夺。因此，设备管理的第一个功能便是按照外部设备的类型和一定的策略（如优先权策略）把外部设备分配给某些作业，与此同时，还应分配相应的通道和控制器。当该作业不再需要某外部设备时，便予以收回。设备管理的第二个功能是启动指定的外部设备进行数据传送操作，和对通道发来的中断请求及时作出响应。

**4. 文件系统** 在较为完善的操作系统中，总是把大量有意义的信息以文件形式存放在各种存储器中，以供所有或指定的用户使用。系统还允许用户把它所输入的信息或计算所得之结果保存在系统中，以便自己或有关用户今后使用。如果在这样的系统中，未配置相应的文件系统，用户存取各种存储器上的信息，就相当繁琐，在多道程序环境时，更是不可想象。为此，在现代操作系统中，都必须配置有文件系统，并提供相应的存储和检索这些信息的手段。

### 1.2.3 虚拟处理机的概念

在单处理机系统中运行多道程序后，从用户观点来看，每个用户都认为系统为自己提供了一台处理机，这样的处理机是逻辑上的，称之为虚处理机。为在一台物理处理机上建立多台虚处理机，应采取下述三点措施：

**1. 分时使用处理机** 多道程序或进程，通过分时方法使用同一台物理处理机。微观地看，该处理机在每个时刻只能运行一道程序；然而宏观地看，一台处理机却在同时运行着几道程序。

**2. 设置管理程序** 为实现多个进程分时使用同一台物理处理机，在系统中必须设置相应的管理程序。它包括用以分配处理机的进程调度程序、分配存储器的内存分配程序、分配I/O设备的设备分配程序，以及用于把用户程序或进程按一定策略排成一个队列的队列管理程序等。每当一进程运行完毕时，进程调度程序便将就绪队列中的第一个进程（如按先来先服务算法）投入运行。利用上述的管理程序，可使各用户进程有条不紊地分时使用一台物理处理机。

**3. 设置进程控制块 PCB 等数据结构** 为实现一台物理处理机的分时使用，在系统中还必须为每个进程设置相应的数据结构，如进程控制块 PCB，它用以记录处理机的状态向量和有关控制、管理信息。每当轮到该进程运行时，便按该进程对应的进程控制块上记录的状态向量来布置物理处理机的现场。在进程运行告一段落后，又把处理机的现场记录在该进