

适用于 IBM PC AT 286 386 PS/2 及其兼容机

# DOS/OS/2操作系统 高级使用技巧

白为民 编译



- 探索DOS核心秘密
- 学习内存常驻程序设计方法
- 学会在高级语言程序中使用DOS系统服务的方法
- 练习开发文件恢复程序
- 附有大量的C或Pascal程序实例
- 完整的源程序代码



中国科学院希望高级电脑技术公司

适用于IBMPCAT 286 386 PS/2及其兼容机

# DOS/OS/2操作系统 高级使用技巧

编译 白为民

中国科学院希望高级电脑技术公司

一九九〇年十月

## 前 言

每天都有数以千计的人开始学习使用DOS。到了今天，约有一千万人已用过DOS，所以DOS可以说是有史以来最具影响力的软件。

市面上一般介绍DOS的书如汗牛充栋，这些书教我们DOS命令的用法，对DOS的内部工作方法也略有提及。虽然这些书关系着我们能不能成功地运用DOS，但是它们的内容也仅限于此，不够充实。本书与这些书大不相同，它将带你跨越这些基本组织，穿过基本命令，来到DOS的核心探索其内部工作方式的深层秘密。在这本书中，你可以试着实际运行大部分的DOS命令，加强许多命令的功能，学会大师级的人才会的内存常驻程序的设计、学会在高级语言程序中使用DOS系统服务的方法，还可以练习开发文件恢复程序。这是第一次，你将进行你在以前只是听说而不敢实际去做的工作。读完本书之后，你会变成一个真正的“DOS顶尖使用者”。

### OS/2与DOS

OS/2是专为80286与80386设计的新的多任务操作系统，DOS每次只执行一个程序，OS/2能允许你同时执行数个程序。身为DOS的“顶尖使用者”，你将比别人先知道OS/2的用法，并可试着为它开发应用软件。本书将为你详细描述OS/2本身及它的能力和操作方法。每章的最后，将简短的说明OS/2对于该章的主题所带来的冲击。事实上，本书也是OS/2的一本完整的入门书。

### 本书的内容

每一章都有一个关于如何成为“顶尖使用者”的主题。在大部分情况下，这些探讨都附有C或Pascal程序，以实例给出该章的主题。本书和其他只讨论理论的书籍的最大不同点在于，本书列出了完整的源程序代码。

本书分为上下册，上册一共九章，其余为下册，每一章节的具体内容是：

第一章展示了顶尖使用者的脑力。本章同时介绍了本书的目标，并简介了OS/2。

第二章介绍了DOS的批处理方式，在这一章你将学会如何使用批参数、具名参数、DOS和OS/2的批处理命令，以及建立循环批处理程序的方法。

第三章介绍了如何通过DOS管道来重新定向I/O的方向，及I/O重新定向运算符有哪些。你也将学会如何设计C和Pascal程序，以提供I/O重新定向功能。此外，你将知道OS/2是怎么增加你的命令行处理能力的。

第四章探讨每个DOS CONFIG·SYS项目的影晌，及磁盘缓冲区和文件头的奥妙。本章对OS/2的每个CONFIG·SYS项目都做详细说明，读完本章之后，相信你对OS/2的多任务工作方式会有所了解。

第五章探讨DOS目录结构，以及它的修改方式、能力和理论基础。你将学会怎么在你的高级语言程序中，执行每一个DOS目录修改命令。

第六章探讨DOS系统调用。DOS提供了一组非常有用的内部程序，称为系统服务程序，可使你在应用程序内调用。在本章中，每项服务都会以高级语言实例给出。这一

章的最后，以OS/2的系统调用作为结束。

第七章介绍几个设计程序的注意事项，这些事项能直接改进你的应用程序的质量。

第八章介绍在DOS和OS/2之下的内存地址确定方法。你将学会将文件放入（或取出）指定的内存地址的方法，本章还将详细的说明OS/2的虚拟存贮。

第九章探讨DOS的命令行处理方式。本章将说明命令参数如何能加强你的程序的功能。如何在C或Pascal程序内存取命令参数，以及DOS将命令行储存在什么地方。如果你的程序是用FORTRAN语言写成的，本章将告诉你设计一个功能强大的命令行分析程序的方法，可供你在FORTRAN语言程序中存取命令行参数。

第十章（也就是下册的第一章）将探讨DOS磁盘的面貌。你将看到磁盘的扇面，磁道和柱面等名词。此外，你将学会读取目录，FAT及系统启动扇面的方法。本章也将告诉你，如何恢复已删除的文件。

第十一章介绍DOS的FORMAT命令，它的能力和运行方式。你会看到几个低层次的ROM BIOS服务程序，可帮助你格式化磁盘上的磁道。

第十二章介绍DOS环境。本章教导你如何存取环境项，并探讨DOS环境能为你的程序带来什么好处。

第十三章介绍DOS的数据结构。你将在检查DOS和OS/2内存图时，学会DOS的日期和时间格式的编码方式，以及什么是启动磁道。

第十四章探讨文件——你可能对文件所做的任何事情。你会设计一些程序，用于修改文件的属性，执行复杂的目录列表，以及扩充每一个DOS文件修改命令的处理能力。本章也将教你如何使用低层次的调用程序，设计出一个完整的ISAM数据库管理系统。

第十五章是最有趣的一章！在这一章内，你将学会如何设计内存常驻程序。本章将告诉你如何捕捉DOS的时钟中断，以便开发更复杂的程序。第十五章中的程序，能够显示很精炼的信息给使用者。

第十六章开始连续四章探讨OS/2。这一章先对OS/2的能力做一番概括性的描述。

第十七章由程序员的观点来看OS/2。在这一章中，你将会建立一个保护模式的程序，并将其移至在相容模式执行的DOS之下。本章详细地探讨了OS/2的API，并给出了几个完整的程序。

第十八章是OS/2的命令参考章节，每个命令都附上完整的例子。

第十九章列出了一系列的问题和解答，以总结对OS/2的探讨。本章对于OS/2做了更详尽的观察，以奠定你的“OS/2脑力”的基础。

附录A是ASC II对照表。

附录B是完整的DOS命令参考对照表。

# 目 录

## 前言

### 第一章 开发顶尖使用者的脑力

- 1.1 DOS的历史..... (1)
- 1.2 学习提问的技巧..... (3)
- 1.3 了解Microsoft Windows所扮演的角色..... (3)
- 1.4 并行概念的发展..... (4)
- 1.5 资源管理..... (5)
- 1.6 共享CPU..... (6)
- 1.7 内存管理..... (10)
- 1.8 OS/2和DOS..... (12)
- 1.9 如何进行准备工作..... (14)

### 第二章 对批处理的研究

- 2.1 基本原理..... (15)
- 2.2 顺序批处理..... (16)
- 2.3 停止批处理..... (18)
- 2.4 基本的批命令..... (19)
- 2.5 条件批处理..... (24)
- 2.6 循环处理..... (27)
- 2.7 参数的使用..... (29)
- 2.8 抽掉装有批处理文件的磁盘的时候..... (32)
- 2.9 改变目录..... (32)
- 2.10 嵌套批处理..... (32)
- 2.11 循环批处理..... (32)
- 2.12 AUTOEXEC、BAT..... (33)
- 2.13 OS/2下的批处理..... (33)
- 2.14 AUTOEXEC、BAT和STARTUP、CMD..... (34)
- 2.15 CMD文件后缀名..... (34)
- 2.16 有名参数..... (34)
- 2.17 DETACH (执行后台命令)..... (35)

### 第三章 DOS的重新定向和管道运算

- 3.1 I/O重新定向..... (37)
- 3.2 DOS的数据管道..... (40)

3.3 用I/O重新定向加强联系.....	(40)
3.4 文件头.....	(41)
3.5 DOS的I/O重新定向设计.....	(41)
3.6 OS/2的I/O重新定向.....	(46)
<b>第四章 DOS系统配置</b>	
4.1 系统配置.....	(41)
4.2 DOS 3.3的参数.....	(64)
4.3 OS/2系统配置参数.....	(68)
<b>第五章 DOS目录</b>	
5.1 介绍DOS子目录.....	(79)
5.2 使用子目录里的文件.....	(89)
5.3 用TREE找文件.....	(91)
5.4 定义执行路径.....	(92)
5.5 用APPEND定义数据文件的路径.....	(94)
5.6 改善系统执行效果.....	(94)
5.7 用磁盘符号代替长路径.....	(95)
5.8 把磁盘连接到目录结构.....	(96)
5.9 注意事项.....	(96)
5.10 应用程序的目录管理.....	(97)
5.11 OS/2的目录管理.....	(97)
<b>第六章 DOS系统服务程序</b>	
6.1 8088寄存器.....	(99)
6.2 INT 21H.....	(100)
6.3 ASCII Z.....	(103)
6.4 其他高级程序语言.....	(158)
6.5 OS/2系统服务程序.....	(162)
<b>第七章 编写DOS应用程序时应注意的事项</b>	
7.1 关键性错误的处理.....	(181)
7.2 可用的磁盘空间.....	(184)
7.3 磁盘名.....	(185)
7.4 出口状态的处理.....	(186)
7.5 建立特殊的文件.....	(189)
7.6 屏幕显示.....	(199)
7.7 文件属性.....	(191)
7.8 CTRL - BREAK的处理.....	(194)
7.9 可供使用的内存空间.....	(194)
7.10 SPAWN程序.....	(195)
7.11 队列式打印.....	(198)

7.12 目标码程序库 .....	(198)
<b>第八章 PC内存地址的确定</b>	
8.1 实际内存 .....	(203)
8.2 内存的映射 .....	(213)
8.3 OS/2的内存地址确定方法 .....	(218)
<b>第九章 加强命令行处理</b>	
9.1 基本命令行处理 .....	(223)
9.2 C的命令行处理 .....	(224)
9.3 Turbo Pascal的命令行处理 .....	(225)
9.4 DOS的命令行处理 .....	(229)
9.5 命令行参数和I/O重新定向 .....	(236)
<b>第十章 磁盘结构</b>	
10.1 磁盘结构 .....	(240)
10.2 扇区、磁道与面 .....	(243)
10.3 有关扇区的一些工作 .....	(245)
10.4 磁盘空间的配置 .....	(250)
10.5 文件分配表 .....	(253)
10.6 目录项 .....	(260)
10.7 文件删除 .....	(269)
10.8 彻底删除 .....	(274)
10.9 读取目录文件 .....	(275)
10.10 其他特性 .....	(282)
10.11 硬盘有什么不同 .....	(286)
10.12 硬盘的结构 .....	(288)
10.13 启动时的工作 .....	(289)
<b>第十一章 DOS的FORMAT命令是如何工作的</b> .....	(291)
<b>第十二章 利用DOS的工作环境</b>	
12.1 检查工作环境 .....	(308)
12.2 利用工作环境项 .....	(315)
12.3 OS/2的工作环境 .....	(317)
<b>第十三章 DOS数据结构分析</b>	
13.1 DOS的内存映射表 .....	(318)
13.2 OS/2的内存映射表 .....	(322)
13.3 启动记录的显示 .....	(323)
13.4 前置程序段 .....	(327)
13.5 介质描述中的信息 .....	(325)
13.6 文件的日期与时间标记 .....	(331)
<b>第十四章 DOS的文件操作</b>	

14.1	文件控制块	(333)
14.2	文件头	(334)
14.3	随机存取文件	(348)
14.4	进一步的DOS文件操作	(373)
14.5	DOS的通配符	(375)
14.6	OS/2的发展方向	(394)
<b>第十五章 内存常驻程序</b>		
15.1	DOS的内存常驻程序	(407)
15.2	扩展键盘预存缓冲区	(408)
15.3	捕获打印屏幕的中断	(410)
15.4	内存常驻时钟程序	(413)
15.5	背景消息处理	(416)
15.6	屏幕存贮	(419)
15.7	警告	(421)
15.8	OS/2的冲击	(421)
<b>第十六章 OS/2 纵观</b>		
16.1	需求	(424)
16.2	多任务	(425)
16.3	真实模式与保护模式	(427)
16.4	虚拟存储器	(427)
16.5	虚拟设备	(431)
16.6	系统需求	(431)
16.7	系统结构	(431)
16.8	进程间的通讯	(433)
16.9	进程调度	(435)
16.10	网络的前景	(436)
16.11	PRESENTATION MANAGER	(436)
<b>第十七章 OS/2 下的程序设计</b>		
17.1	一个简单的例子	(438)
17.2	BIND	(439)
17.3	EXE结构	(439)
17.4	模块定义文件	(442)
17.5	BIND的处理能力	(445)
17.6	MAKE	(445)
17.7	OS/2 应用	(447)
<b>第十八章 OS/2 命令汇总</b>		
18.1	OS/2 的作业	(455)
18.2	CMD·EXE与COMMAND·COM	(456)



18.3 OS/2 命令 .....	(458)
<b>第十九章 有关OS/2的问题与解答</b>	
19.1 了解需求 .....	(487)
19.2 正式开始 .....	(488)
19.3 哲学 .....	(489)
19.4 系统说明 .....	(491)
19.5 一般问题 .....	(502)

## 第一章 开发顶尖使用者的脑力

DOS的顶尖使用者有共同的愿望。他们在了解了DOS的基本用法之后，都会产生一种渴望，希望能够获得控制他们与电脑系统及应用软件之间界面的软件能力。DOS的顶尖使用者充满了好奇心，他们想知道DOS的基本运行原理，以及如何加快DOS的运行速度。大部分的DOS顶尖使用者是程序设计大师，但也不全是这样。有许多事业有成的顾问，只因为对DOS的内部流程有深刻的了解，就能够将工具软件和硬件结合在一起，从而成为他们那一行的顶尖人物。不论是想追求财实、成就、知识或只是兴趣所在，这些DOS顶尖使用者都有着相同的目标：希望对于手边的DOS能有完整而深刻的了解。

本章所要安排的素材，足够让你变成一个DOS顶尖使用者。此外，本章还详述了DOS自1981年诞生以来完整的发展史；版本修改的原因和在发展过程中取得的经验。

本章也将引导你踏入OS/2之门。作为一个DOS的顶尖使用者，你将走在微电脑历史的前端。OS/2虽是尚未完全成熟的作品，但是它所迈出的这一步，已将微电脑由家用电脑的角色，带进入全功能、多任务工作站领域中。本章将告诉你OS/2的发展目标，运行情况及其他相关事项。请记住，你曾花费许多努力去研究DOS，并已完全了解它的内部工作方式，现在该是对OS/2做同样的事情的时候了。

本章最后将展望DOS未来的命运。究竟DOS的未来如何，OS/2会给使用者、开发者、程序员及数据处理管理带来什么样的冲击？你的下一步该怎么走？你又该做什么样的准备，以迎接微电脑的未来。

### 1.1 DOS的历史

在1981年IBM-PC推出时，IBM和Microsoft都推出了1.0版的DOS（分别是PC-DOS和IMS-DOS），这个原始的版本只提供了一些基本的命令给使用者。但重要的是，它为今天的DOS和OS/2建立了基本的文件管理系统。这个文件管理系统是建立在文件配置表（FAT）上。自1.0版本之后的DOS，都具备了文件的相容性，换句话说，DOS 3.x和OS/2仍然能读取由DOS 1.0所产生的文件。虽然DOS 1.0所提供的命令集合就今天的操作系统的观点来看，实在是少得可怜，但是DOS 1.0的设计者所表现的预见能力，确实令人折服。

1982年，Microsoft对DOS做了一番加强，提供了双面磁盘的存取能力。早期的PC使用者应该还记得，那时PC使用的是单面磁盘，只能贮存160K的文件。

1983年，IBM和Microsoft都推出了DOS2.0。就在这时，DOS决定追随UNIX的功能，跳出CP/M的窠臼。DOS2.0完全提供了DOS子目录DOS管道及一组加强的命令集合。此外DOS2.0提供了由文件控制块（FCB）转移至文件头的功能，从而实现了I/O重新定向功能。

在1983年中，IBM和Microsoft对于DOS做了一些加强性的工作，重点在于改正

系统的错误。或许你对于DOS版本号来源不甚清楚，现在在此做个简短说明。每个DOS版本号可以分解为两部分，一个是主版本号，另一个是次版本号。例如，DOS 3.2中的3是主版本号，而2是次版本号。

大部分的软件开发公司，都采取一种习惯用法来设定修改版本号：如果对软件做了大的修改，主版本号就递增（例如由2.0变成3.0）。如果所做的改变只是次要的改变（例如修正一个错误），就只递增次版本号。因此，1983年时，共推出了2.0、2.01、2.10、2.11和2.25版的DOS。

到了1984年，IBMPCAT诞生了，这时候Microsoft和IBM都推出了DOS 3.0。DOS 3.0提供了1.2MB软盘和容量更大的硬盘。就在同一年，两家公司又推出了DOS 3.1，具备支持局域网的能力。DOS 3.1在功能上的改变并不大，但是它却改变了DOS的面貌，并建立了多用户和工作站的基础。由8年前的单用户系统开始，在短短三年内，DOS已具备足够的功能；在多用户领域内向UNIX和XENIX挑战。

1986年诞生了便携式电脑。这又是另一次强烈的冲击。为了加大磁盘的容量，减少磁盘占用的空间，便携式电脑改用3.5吋的磁盘。IBM和Microsoft都推出了能提供3.5吋微软磁盘的DOS3.2。

1987年4月，IBM推出了个人系统/2（PS/2）系列电脑。这个系列的电脑提供了新的技术，以加强8088、80286和80386等微处理机的功能。在这同时，IBM推出了DOS 3.3，以供PS/2系列使用，而Microsoft和IBM也都推出了OS/2，这是专供80286和80386微处理机使用的多道程序环境。本书将在本章及以后的章节中详细解释OS/2。

表1-1和1-2摘要列出了自1981年起DOS的发展过程。

表1-1 DOS的版本

版本	日期	功 能
1.0	1981	原始的磁盘操作系统
1.25	1982	提供了双面磁盘
2.0	1983	提供了子目录
2.01	1983	提供了国别符号
2.11	1983	修改错误
2.25	1983	提供了扩充字符集
3.0	1984	提供了1.2MB软盘，大型硬盘
3.1	1984	提供了PC网络
3.2	1986	提供了微型软盘
3.3	1987	可供IBM PS/2电脑使用
OS/2	1987	全功能的多任务环境

表1-2 IBM的DOS

版本	日期	功 能
1.0	1981	原始的磁盘操作系统
2.0	1983	提供了子目录
2.10	1983	修改错误
3.0	1984	提供了1.2MB软盘, 大型硬盘
3.1	1984	提供了PC网络
3.2	1986	提供了微型软盘
3.3	1987	可供IBM PS/2电脑使用
OS/2	1987	全功能多任务环境

### 1.2 学习提问的技巧

在学习过程中不断地提出问题, 对于提高学习效果有很大帮助, 你阅读DOS的历史时, 最好能列出下列问题:

叙述: 1983年DOS 2.0引进了文件头。

问题: DOS使用文件头后带来了I/O重新定向功能了吗? 如果是, DOS是怎么做的。

问题: 为什么不用文件控制块? 它有什么缺点?

问题: 文件头和CONFIG.SYS文件内的HANDLES = 项目是否存在着对应关系?

问题: 你所设计的应用软件如何去存取文件头?

所能列出的问题可能无止境, 所以最好准备一本笔记本将问题全记下来。

本书将为你解答大部分的问题。更重要的一点是, 本书还为你发掘出问题。只有透过提问并寻求解答, 才能获得真正的了解, 这是顶尖学习者的学习要诀。

### 1.3 了解Microsoft Windows所扮演的角色

如果将DOS视为80年代最具影响的系统软件, 那么Windows可以说是仅次于DOS的系统软件。如果你还没用过Windows, 千万不要错过; 如果你曾用过Windows, 但是并不喜欢它, 可否请你再慎重考虑一次。OS/2提供了一套软件界面, 称为Presentation Manager, 这个程序负责将OS/2呈现在使用者的面前, 但OS/2的Presentation Manager所提供的界面及它的资源控制方式, 基本构想来自于Windows。在可见的未来, 大部分执行OS/2的PC, 都会采用这样的界面。所以不论你喜欢不喜欢, 这已是一种事实, 更何况Presentation Manager有许多的优点。

Presentation Manager在最高层次为所有应用软件提供了具有一致性的使用界面, 所以使用者不须记忆DOS命令。惯于使用命令行的用户无须惊慌, OS/2仍允许你使用命令行, 但是因为大多数的使用者都将改用Presentation Manager, 所以你最好还是跟随潮流。因此, 从现在开始使用Windows吧!

对于软件开发者而言，在OS/2下设计程序和Windows下设计程序没有两样。如果你希望所开发的应用软件到了1990年仍不淘汰，最好拥有Windows的软件开发卡，并开始学习在Presentation Manager之下开发应用软件的方法。以下列出一本绝佳的参考书：Windows Programming Secrets, Kris Jamsa著 (Berkeley, Calif, Osborne/McGraw-Hill, 1987)

#### 1.4 并行概念的发展

到目前为止，大部分的人都习惯于在DOS出现提示符之后，跟着输入命令。进入DOS后，DOS会送出提示符，要求你输入命令。然后你输入命令，DOS执行命令。这种处理方式不断重复，直到你关掉电脑为止。这种顺序的命令执行方式（一个接着一个执行命令），对于大部分的使用者而言，是他们执行DOS命令的唯一方式。他们已习惯于电脑每次只执行一条命令；但是，DOS的PRINT命令却采取不同的工作方式。当文件放入打印队列时，DOS却能够一边执行其他命令，一边打印文件。

个人电脑只有一个CPU。所以每次只执行一件工作是件很自然的事。但是多任务操作系统的目标，却是希望让几件工作看起来好象是在同时执行，这就有赖于CPU的分时了。

举个例子，假设你正使用正文处理程序输入英文字母，因为CPU的速度太快，因此它的大部分时间花在等你输入下一个字母上。这个等待时间，或称为CPU的闲置时间，会占据CPU使用时间的一大部分。（见图1-1）

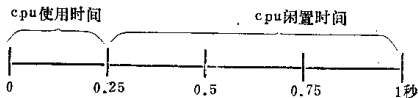


图1-1 CPU处理时间

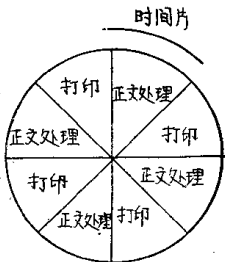


图1-2 CPU处理时间举例

为了让CPU获得充分的利用，多任务操作系统将CPU的时间分割成小片断，称为时间片，CPU将时间片分配给每个运行中的程序。例如，假设你用DOS PRINT命令打印文件的内容时，同时也在使用正文处理程序，CPU时间的分割情形会象图1-3一般。

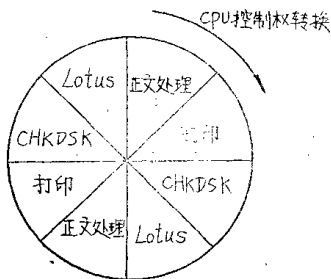


图1-3 CPU时间共享

每个程序获得一小段时间的CPU控制权，而利用这一小段时间进行它们的工作。操作系统会在时间到时，将CPU的控制权转移给下一个程序。在本例中，正文处理程序获得CPU的初始控制权，然后是PRINT命令获得下一个时间片断的CPU控制权。接着由正文处理程序获得控制权，如此无限循环下去。因为这种转换异常快速，所以这两个程序会给人同时执行的印象。当OS/2开始执行更多的程序时，每个程序会分配到一个时间片（见图1-3）。这样，操作系统看起来就象是在执行多个程序一样。

### 1.5 资源管理

在多道程序环境中，操作系统的主要功能之一就是资源管理，所谓资源管理，就是要决定程序是否能存取特定的硬件设备，及何时存取。例如，假设有两个程序正在并行运行，其中一个正在屏幕上显示其执行结果时，另一个程序不可在此时将其输出显示在屏幕上。否则，屏幕上同时显示两个程序的输出，将会显得杂乱无章。同理，读者也能想象多个程序同时将数据写入磁盘时所导致的混乱情形。

在同一个环境下执行的多道程序必须互相合作。OS/2含有数个子系统，负责确保资源分配的一致性。这些子系统共同组成了OS/2的资源管理系统。当然，不是只有OS/2才有资源管理系统，所有的多道程序环境，例如，Windows, Topview或OS/2，都有完整的资源管理功能。

### 1.6 共享CPU

OS/2 是个分时操作系统，它将CPU的处理时间平均分配给系统中的每个程序。靠着这种方式，OS/2 给予使用者同时执行多个程序的印象。在大部分的情况下，OS/2 是以循环法分配CPU的时间的。

但是，要是并行执行的程序数量块多后，程序等待接受CPU时间的时间也会相对的增加。

这就是当你在Windows之下装入多个程序时，系统中所有的程序似乎变成静止不动的原因。造成反应速度缓慢的第二个原因是内存的共享（文件移动）问题。虽然以循环法分配CPU时间是种最公平的办法，但就实际的观点来看，当程序所需的数据已准备好时，程序就必须能够使用CPU了。

请考虑下面这个情况，你的多道程序系统正执行8个程序，第一个程序是考勤程序，记录职工的工作进度、出席情况及工作成果。第二个程序是正文处理程序，用于产生周报表。第三个程序是核子反应炉的销毁监督程序（见图（1-4））。

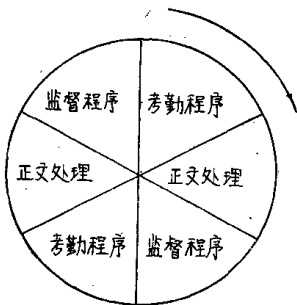


图 1-4 CPU时间片的分布情形

在时间n时，监督程序收到下列数据：

```
Time message  
.....  
n Possible Problem in Sector eight
```

监督程序要求反映炉系统提供更多的信息。同时，操作系统将控制权转移至考勤程序，以运行这个程序（见图1-5）。

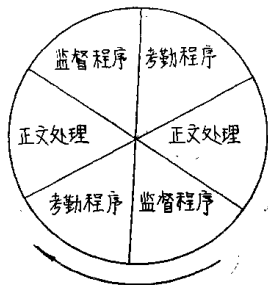


图 1 - 5 考勤程序取得CPU的控制权

接着轮到正文处理程序执行（见图 1 - 6）。最后CPU的控制权又回到监督程序（见图 1 - 7）。此时监督程序收到下列消息：

```
time message
.....
n+3 please advise
```

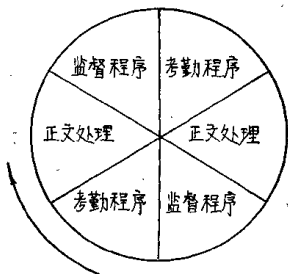


图 1 - 6 正文处理程序取得CPU控制权



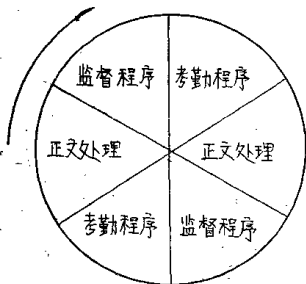


图 1 - 7 监督程序取得CPU控制权

这个消息令人不解，反应炉究竟要监督程序做什么？原来，当监督程序所要求的警告信息到达时，监督程序并未处于执行状态，所以消息丢失了！这时，监督程序只好再度要求反应炉提供信息。但是，它仍须放弃CPU的控制权（参看图 1 - 8）。下次CPU再执行监督程序时，它收到下列的消息：

```
Time message
.....
n+6 Perform reactor Shutdown (Default, no)
```

但是在它准备要回应此消息时，CPU时间片断已做完，因此只好等到下一次。但是，反应炉已不耐烦等而自行销毁。因此，下次监督程序执行时，它会收到下面的消息：

```
Time message
.....
n+9 meltdown complete. No message to follow
```

很明显，循环法确定顺序的方法必须有所改进才行。OS/2 对调度算法所做的改进是为系统中的每个程序设定一个优先顺序。时间紧迫的应用（如数据通讯或是原子反应炉监督程序），优先次序高于一般的应用（如正文处理程序）。

每当OS/2 执行完一个时间片后，就检查它的高优先顺序的程序队列，看看里头是否含有可以执行的程序。如果有，则执行这个程序；如果没有，OS/2 对优先次序稍低的队列做相同的搜寻工作。以上一个例子为例，监督程序的优先权你可以给它 2，而另两个程序则可以给它 1。（见图 1 - 9）。

某一个时间片断执行完之后，OS/2 先检查监督程序，程序如果准备好了，就加以执行，否则对其余程序进行循环分配。见图 1 - 10流程图。