

面向21世纪计算机专业本科系列教材

操作系统原理

何炎祥 熊前兴 编著

华中科技大学出版社
(华中理工大学出版社)

图书在版编目(CIP)数据

操作系统原理/何炎祥 熊前兴 编著
武汉:华中科技大学出版社, 2001 年 10 月
ISBN 7-5609-2561-8

I. 操…

II. ①何… ②熊…

III. 操作系统-高等学校-教材

IV. T316

操作系统原理

何炎祥 熊前兴 编著

责任编辑:沈旭日

封面设计:刘 卉

责任校对:章 红

责任监印:熊庆玉

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87545012

经 销:新华书店湖北发行所

录 排:华中科技大学惠友文印中心

印 刷:核工业三〇九印刷厂

开本:787×960 1/16

印张:24

字数:430 000

版次:2001 年 10 月第 1 版

印次:2001 年 10 月第 1 次印刷

印数:1—5 000

ISBN 7-5609-2561-8/TP·444

定价:27.50 元

(本书若有印装质量问题,请向出版社发行部调换)

前 言

计算机系统存在着极大的多样性，不仅机型、容量和速度存在差异，而且应用程序和系统支持环境也不尽相同；此外，整个计算机系统正以极快的速度发展着，作为计算机系统中的十分关键的系统软件——操作系统，也不断有新的关键技术诞生。因此，本书将结合当今操作系统的设计并考虑操作系统的发展方向，着重讨论操作系统设计的基本概念、基本原理和典型技术，同时，介绍构造操作系统过程中可能面临的种种问题及其解决办法；介绍操作系统设计中的一些非常重要的新进展，包括线程、实时系统、多处理器调度、分布式系统、中间件、进程迁移和安全性等。其目的是尽可能清晰、全面地向读者展现当代操作系统的设计原理与基本实现技术，以便读者深入了解现代操作系统，为今后进行较深层次的软件研制与系统开发打下坚实的基础。

为了帮助读者更好地理解操作系统的概念、原理和方法，更好地将理论与实际设计相结合，我们选择了以下3个具有相关性、代表性和典型性的操作系统作为例子。

- Windows NT：一个单用户、多任务操作系统，第一个采用面向对象设计原理设计的重要操作系统。它运行在个人计算机、工作站和服务器的上。

- UNIX：一个多用户、多任务、分时交互式操作系统。它开始是为微型计算机设计的，现在已广泛应用于其他机型。

- MVS：IBM大型计算机的顶级操作系统，是到目前为止最为复杂的操作系统。它提供了批处理和分时处理的功能。

本书由以下12章组成。

第1章操作系统概述：对本书的内容进行了概述。

第2章进程的描述和控制：讨论了进程的概念以及操作系统对进程进行控制的数据结构，还讨论了与进程相关的线程等内容。

第3章并行控制——互斥和同步：着重讨论在单一系统中并行处理的关键技术——互斥和同步机制。

第4章死锁处理：描述了死锁和饥饿的性质并讨论了解决它们的方法。

第5章内存管理：提出了多种内存管理方法，并讨论了支撑虚拟内存所需的硬件结构和操作系统用来管理虚拟内存的软件方法。

第6章处理机调度：讨论了各种不同的进程调度方法，包括实时调度策略。

第7章I/O管理和磁盘调度：论述了操作系统对输入/输出的控制，尤其是对系统性能影响较大的磁盘I/O的调度和控制。

第8章文件管理：对文件的组织、存储、使用和保护等方面的内容进行了综述。

第9、10章分布计算和分布式进程管理：描述了分布式操作系统的一些关键设计领域，包括Client/Server结构，用于消息传递和远程过程调用的分布式通信机制、分布式进程迁移、中间件以及解决分布式互斥和死锁问题的原理与技术。

第11章操作系统的安全性：简要讨论了保证计算机系统和网络安全性的相关理论和方法。

第12章排队分析理论：扼要介绍了排队分析的基本原理，及其在操作系统设计中的应用。

本书是参照全国高校计算机专业教学指导委员会、中国计算机学会教育委员会和全国高等学校计算机教育研究会联合推出的《计算机学科教学计划2000》，从传授知识和培养能力的目标出发，结合作者多年从事教学与科研的实践，根据本课程教学的特点、重点和难点编写的。本课程的先导课程是“高级语言程序设计”，“数据结构”，“计算机组成原理”等。本课程的教学参考学时为72学时，也可根据需要对其内容进行取舍，将学时压缩到54学时。

为了满足教学和自学的需要，我们还编写了配套教材《操作系统原理学习与解题指南》，供读者选用。

本书的第1~4，6，10，12章由何炎祥编写，第5，7~9，11章由熊前兴编写，何炎祥统编了全书。在本书的编写过程中，得到了武汉大学计算机学院和武汉理工大学计算机学院的领导和同事们的关心，华中科技大学出版社为本书的出版给予了大力支持，文中还引用了一些专家学者的研究成果和公司的产品介绍，在此一并表示感谢。

限于水平，书中错误难免，敬请读者赐教。

编著者

2001年7月于武昌珞珈山

目 录

第 1 章 操作系统概述	(1)
1.1 操作系统的作用	(1)
1.1.1 作为人机交互界面	(1)
1.1.2 作为资源管理者	(2)
1.1.3 推动操作系统发展的因素	(3)
1.2 操作系统的演变	(4)
1.2.1 串行处理系统	(4)
1.2.2 简单批处理系统	(4)
1.2.3 多道程序批处理系统	(7)
1.2.4 分时系统	(9)
1.3 操作系统的主要成就	(11)
1.3.1 进程	(11)
1.3.2 存储器管理	(14)
1.3.3 信息保护和安全性	(15)
1.3.4 调度和资源管理	(16)
1.3.5 系统结构	(17)
1.4 操作系统举例	(19)
1.4.1 Windows NT	(19)
1.4.2 UNIX System V	(23)
1.4.3 MVS	(25)
1.5 操作系统的主要研究课题	(27)
习题一	(27)
第 2 章 进程描述与控制	(29)
2.1 进程状态	(29)
2.1.1 进程产生和终止	(31)
2.1.2 进程状态模型	(33)
2.1.3 进程挂起	(36)

2.2	进程描述	(40)
2.2.1	操作系统控制结构	(41)
2.2.2	进程控制结构	(41)
2.2.3	进程属性	(42)
2.3	进程控制	(44)
2.3.1	执行模式	(44)
2.3.2	进程创建	(45)
2.3.3	进程切换	(45)
2.3.4	上下文切换	(46)
2.3.5	操作系统的运行	(47)
2.3.6	微核	(48)
2.4	线程和 SMP	(49)
2.4.1	线程及其管理	(49)
2.4.2	多线程的实现	(51)
2.4.3	进程与线程的关系	(52)
2.4.4	SMP	(53)
2.5	系统举例	(54)
2.5.1	UNIX System V	(54)
2.5.2	Windows NT	(58)
2.5.3	MVS	(62)
2.6	小 结	(65)
	习题二	(65)
第 3 章	并发控制——互斥与同步	(68)
3.1	并发原理	(69)
3.1.1	进程间的相互作用	(71)
3.1.2	进程间的相互竞争	(72)
3.1.3	进程间的相互合作	(74)
3.1.4	互斥的要求	(75)
3.2	互斥——用软件方法实现	(75)
3.2.1	Dekker 算法	(76)
3.2.2	Peterson 算法	(79)
3.3	互斥——用硬件方法解决	(80)
3.3.1	禁止中断	(80)
3.3.2	使用机器指令	(81)

3.4 信号量	(83)
3.4.1 用信号量解决互斥问题	(84)
3.4.2 用信号量解决生产者/消费者问题	(85)
3.4.3 信号量的实现	(89)
3.4.4 用信号量解决理发店问题	(90)
3.5 管程	(94)
3.5.1 带信号量的管程	(94)
3.5.2 用管程解决生产者/消费者问题	(95)
3.6 消息传递	(97)
3.6.1 消息传递原语	(97)
3.6.2 用消息传递实现同步	(98)
3.6.3 寻址方式	(98)
3.6.4 消息格式	(99)
3.6.5 排队规则	(100)
3.6.6 用消息传递实现互斥	(100)
3.7 读者/写者问题	(101)
3.7.1 读者优先	(102)
3.7.2 写者优先	(103)
3.8 小 结	(105)
习题三	(106)
第 4 章 死锁处理	(108)
4.1 死锁概述	(108)
4.1.1 可重用资源	(108)
4.1.2 消耗型资源	(109)
4.1.3 产生死锁的条件	(110)
4.2 死锁处理	(111)
4.2.1 死锁预防	(111)
4.2.2 死锁检测	(112)
4.2.3 死锁避免	(112)
4.2.4 采用综合方法处理死锁	(116)
4.3 哲学家用餐问题	(117)
4.4 系统举例	(118)
4.4.1 UNIX System V	(118)
4.4.2 Windows NT	(120)

4.4.3	MVS.....	(121)
4.5	小 结	(123)
	习题四	(123)
第 5 章	内存管理.....	(125)
5.1	概 述	(125)
5.1.1	基本概念	(125)
5.1.2	虚拟存储器	(127)
5.1.3	重定位	(128)
5.2	存储管理的基本技术	(130)
5.2.1	分区法	(130)
5.2.2	可重定位分区法	(132)
5.2.3	覆盖技术	(134)
5.2.4	交换技术	(134)
5.3	分页存储管理	(135)
5.3.1	基本概念	(135)
5.3.2	纯分页系统	(138)
5.3.3	请求式分页系统	(140)
5.3.4	硬件支持及缺页处理	(140)
5.3.5	页的共享和保护	(141)
5.4	分段存储管理	(142)
5.4.1	基本概念	(143)
5.4.2	基本原理	(143)
5.4.3	硬件支持和缺段处理	(144)
5.4.4	段的共享和保护	(146)
5.5	段页式存储管理	(147)
5.5.1	基本概念	(147)
5.5.2	地址转换	(148)
5.5.3	管理算法	(149)
5.6	虚拟内存的置换算法	(150)
5.6.1	先进先出页面置换算法	(150)
5.6.2	最佳页面置换算法	(151)
5.6.3	最近最少使用页面置换算法	(151)
5.6.4	第二次机会页面置换算法	(152)
5.6.5	时钟页面置换算法	(153)

5.6.6 其他页面置换算法	(153)
5.7 系统举例	(154)
5.7.1 UNIX 系统中的存储管理技术	(154)
5.7.2 Linux 系统中的存储管理技术	(158)
5.8 小 结	(160)
习题五	(161)
第 6 章 处理机调度	(162)
6.1 调度类型	(162)
6.1.1 长程调度	(164)
6.1.2 中程调度	(164)
6.1.3 短程调度	(164)
6.2 调度算法	(164)
6.2.1 短程调度标准	(164)
6.2.2 优先权的使用	(166)
6.2.3 调度策略	(166)
6.2.4 性能比较	(174)
6.2.5 模拟模型	(176)
6.2.6 公平分享调度策略	(177)
6.3 多处理机调度	(179)
6.3.1 粒度	(179)
6.3.2 设计要点	(180)
6.3.3 进程调度策略	(182)
6.4 实时调度	(186)
6.4.1 实时操作系统的特性	(187)
6.4.2 实时调度	(189)
6.4.3 期限调度	(190)
6.4.4 比率单调调度	(193)
6.5 系统举例	(195)
6.5.1 UNIX System V	(195)
6.5.2 Windows NT	(197)
6.5.3 MVS	(198)
6.6 小 结	(200)
附录 响应时间	(200)
习题六	(201)

第 7 章 I/O 设备管理	(204)
7.1 I/O 系统硬件	(204)
7.1.1 I/O 设备	(204)
7.1.2 设备控制器	(205)
7.1.3 I/O 技术	(207)
7.2 I/O 软件	(212)
7.2.1 中断处理程序	(212)
7.2.2 设备驱动程序	(213)
7.2.3 与设备无关的 I/O 软件	(214)
7.2.4 用户空间的 I/O 软件	(216)
7.2.5 缓冲技术	(217)
7.3 磁盘调度	(222)
7.3.1 调度策略	(222)
7.3.2 磁盘高速缓存	(225)
7.4 系统举例	(227)
7.4.1 UNIX System V	(227)
7.4.2 Windows NT I/O 分析	(229)
7.5 小 结	(233)
习题七	(234)
第 8 章 文件管理	(236)
8.1 文件与文件系统	(236)
8.1.1 文件及其分类	(236)
8.1.2 文件系统及其功能	(237)
8.2 文件的结构及存取方式	(239)
8.2.1 文件的逻辑结构及存取方式	(239)
8.2.2 文件的物理结构及存储设备	(241)
8.3 文件管理	(247)
8.3.1 文件目录结构	(247)
8.3.2 文件目录管理	(252)
8.4 文件存储空间的分配与管理	(253)
8.4.1 文件存储空间的分配	(253)
8.4.2 磁盘空间管理	(255)
8.5 系统举例——Windows NT	(258)

8.5.1 PE 可移动执行的文件格式	(258)
8.5.2 PE 文件首部	(260)
8.5.3 块表数据结构及辅助信息块	(263)
8.6 小 结	(268)
习题八	(268)
第 9 章 分布计算	(269)
9.1 客户/服务器计算	(270)
9.1.1 什么是客户/服务器计算	(270)
9.1.2 客户/服务器模式的应用	(271)
9.1.3 中间件	(275)
9.2 分布式消息传递	(277)
9.2.1 分布式消息传递的方法	(277)
9.2.2 消息传递的可靠性	(278)
9.3 远程过程调用	(278)
9.4 小 结	(280)
习题九	(280)
第 10 章 分布式进程管理	(281)
10.1 进程迁移	(281)
10.1.1 进程迁移的动机	(281)
10.1.2 进程迁移的机制	(282)
10.1.3 进程迁移的协商	(284)
10.1.4 进程驱逐	(285)
10.1.5 抢占及非抢占进程的迁移	(286)
10.2 分布式全局状态	(286)
10.2.1 全局状态及分布式快照	(286)
10.2.2 分布式快照算法	(288)
10.3 分布式进程管理——互斥	(290)
10.3.1 分布式互斥	(290)
10.3.2 分布式系统的事件定序——时戳方法	(292)
10.3.3 分布式互斥算法	(294)
10.4 分布式死锁	(298)
10.4.1 资源分配中的死锁	(298)
10.4.2 死锁预防	(299)

10.4.3 死锁避免.....	(300)
10.4.4 死锁检测.....	(300)
10.4.5 消息通信中的死锁.....	(303)
10.5 小 结	(306)
习题十	(306)
第 11 章 操作系统的安全性	(308)
11.1 安全性概述.....	(308)
11.1.1 安全性的内涵	(308)
11.1.2 操作系统的安全性	(309)
11.1.3 安全性级别	(312)
11.2 安全保护机制.....	(313)
11.2.1 进程支持.....	(313)
11.2.2 内存及地址保护.....	(314)
11.2.3 存取控制.....	(317)
11.2.4 文件保护	(321)
11.2.5 用户身份鉴别	(324)
11.3 病毒及其防御.....	(326)
11.3.1 病毒概述.....	(326)
11.3.2 病毒的防御机制.....	(327)
11.3.3 特洛伊木马程序及其防御.....	(329)
11.4 加密技术.....	(329)
11.4.1 传统加密方法	(330)
11.4.2 公开密钥加密方法	(331)
11.4.3 密钥的管理	(332)
11.5 安全操作系统的设计.....	(334)
11.5.1 安全模型.....	(334)
11.5.2 安全操作系统的设计.....	(337)
11.6 系统举例——Windows 2000 的安全性分析.....	(341)
11.7 小 结	(344)
习题十一	(344)
第 12 章 排队分析理论	(346)
12.1 为什么进行排队分析	(346)
12.2 排队模型	(347)

12.2.1 单服务器模型	(347)
12.2.2 多服务器模型	(349)
12.2.3 基本排队关系	(349)
12.2.4 假设	(349)
12.3 单服务器队列	(350)
12.4 多服务器队列	(351)
12.5 队列网	(352)
12.5.1 信息流的分割和汇聚	(353)
12.5.2 一前一后的队列	(353)
12.5.3 Jackson 定理	(353)
12.5.4 包交换网中的应用	(354)
12.6 其他排队模型	(355)
12.7 小 结	(355)
参考文献	(356)



操作系统概述

操作系统(Operating Systems, 简称 OS)是控制应用程序执行和充当人机交互界面的软件。OS 常可以实现 3 个目标和执行 3 种功能:

- ① 方便。OS 使计算机应用起来更方便。
- ② 高效。OS 使计算机系统的资源以一种高效的形式得以应用。
- ③ 可扩展。建造 OS 时, 应使它具有进行扩充和发展, 并且在引进新系统组件的同时, 又不会干扰现有服务的能力。

1.1 操作系统的作用

1.1.1 作为人机交互界面

如图 1.1 描述的一样, 计算机用以给用户提供应用的软、硬件可看做是层次化、分级的形式。终端用户通常不必考虑计算机本身的设计, 只是借助应用程序与计算机系统交互。这些应用程序可通过程序设计语言来表达, 并且可被应用程序员升级。如果一个人想通过一系列机器指令来开发应用程序, 而这些应用程序又全权控制计算机硬件, 那么, 等待他的将是一个十分庞杂的工作。为简化这项工作, 一系列系统程序则应运而生。在这些系统程序中, 有些被称为实用程序, 它们频繁地执行程序创建、文件管理、输入/输出(I/O)设备控制等功能。程序员可以利用它们来开发应用程序, 而应用程序在运行时又调用实用程序执行某项功能。最重要的系统程序是 OS。由 OS 定义的软、硬件和数据,

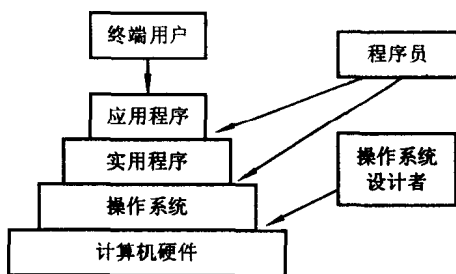


图 1.1 计算机系统的层次视图

给程序员提供了方便的界面，使程序员和应用程序更容易获取和使用计算机系统内的资源、工具和服务。

简言之，OS 为如下领域提供服务：

① 程序创建。OS 提供多种工具（如编辑器、调试器）和服务来帮助程序员编程。特别要说明的是，这些服务以实用程序的形式存在，它们并不是 OS 的一部分，而只是通过 OS 获得。

② 程序执行。将指令和数据装入主存、I/O 设备和文件初始化，以及其他资源的准备等，这些工作是由 OS 完成的。

③ I/O 设备的访问。每个 I/O 设备都有用来操作自身的指令系列和控制信号，OS 则可使程序员用简单的读/写操作来使用和控制 I/O。

④ 控制对文件的访问。就文件而言，控制不仅能识别 I/O 设备属性(磁盘驱动器，打印驱动器)，而且能识别存储介质上的文件结构。当涉及到多用户系统时，OS 能提供控制访问文件的保护机制。

⑤ 系统访问。当涉及一个共享或公用的系统时，OS 承担整体控制系统的访问和具体系统资源的访问。它必须使资源和数据不受无权用户的干扰，而且必须解决资源使用中的冲突问题。

⑥ 查错和纠错。大量的错误不能在系统运行时出现，包括内部和外部硬件错误，例如存储器错误，或设备失误、无效；各种软件错误，例如，算术溢出、试图使用禁用存储器，OS 不能满足应用程序要求等。每种情况下，OS 都必须作出响应，并在最小限度影响执行程序的前提下清除错误，或向应用程序报错，或重试一次，或终止出错程序。

⑦ 簿记。好的 OS 应能搜集各种资源的使用统计数据 and 监听执行情况(如响应时间)等。这些信息对将来完善计算机所需的条件，以及提高系统执行性能是十分有用的。

1.1.2 作为资源管理者

是否能说 OS 控制运行、存储和处理数据呢？从某种观点来看是对的：OS 通过管理计算机资源来控制计算机基本功能的实现。但这个控制是以不寻常的方式运作的。通常认为一个控制部件应位于其被控物的外部，或至少是不同于且分离于被控物的，OS 就不是这样，作为一个控制部件它至少有两个方面不寻常：

- 以通常的计算机软件的方式起作用，即它是一些程序的集合。
- 频繁放弃控制，且必须依靠处理器来重新获取控制。

OS 实际上只不过是一个计算机程序。它与一般程序的关键区别是程序的内容。OS 可指示处理器使用其他系统资源，加速其他程序的执行。但这时，处理器必须停止执行 OS 程序。这样，OS 就会为处理器而放弃控制，去做另一些“有用”工

作，待到重新掌握控制权后再为处理器的下一项工作做准备。

图 1.2 显示出一些受 OS 控制的主要资源。OS 的一部分在内存中。包括内核，它包含了 OS 中使用最频繁的和在给定时间内正被使用的 OS 的一部分。内存余下的部分存储了其他用户程序和数据。主存的分配是由 OS 和处理器中的存储管理硬件决定的，如图 1.2 所示，OS 决定什么时间 I/O 设备能够被处于执行中的程序段用到，还控制和管理对文件的使用。处理器本身是资源，OS 决定处理器有多少时间花在执行特定用户程序上。

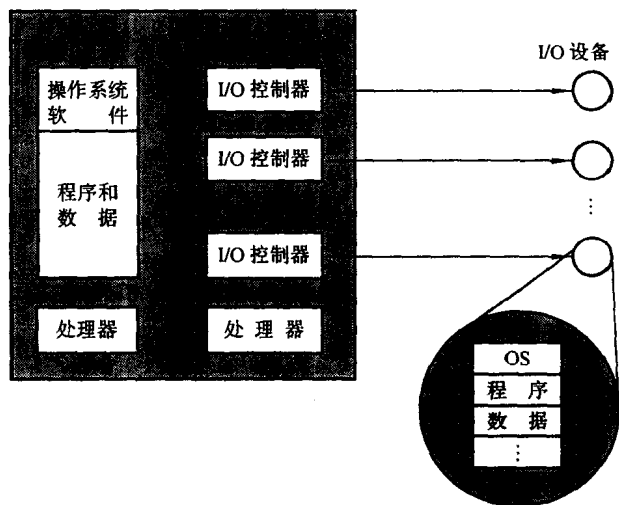


图 1.2 作为资源管理器的操作系统

1.1.3 推动操作系统发展的因素

操作系统随时间而演变主要基于以下因素。

1. 硬件升级以及新的硬件类型

例如，早期的 UNIX 和 OS/2 不含分页部件，因它们的运行不需分页硬件，最近的版本已增加了分页能力；同样，图形终端和页式终端代替逐行显示的终端会影响 OS 的设计，这样的终端允许用户通过屏幕上的窗口，在同一时间内提出多个申请，这就需要 OS 有更多更复杂的支持。

2. 新服务

为响应用户要求和系统管理者的需要，OS 要扩大服务范围。当发现现有的工具很难为用户保持良好的服务性能时，将新的测量和管理工具添加到 OS 中；在出现一个新请求，如要求在显示器上应用 Windows 操作系统时，就需要有相应的支撑机制。

3. 修补

OS 中的错误会不断被发现, 因此就必须进行修补。当然, 在修补的同时也有可能引入新的错误。

显而易见, OS 在创建中应以组件为基础, 应具有清晰定义的各组件之间的接口, 应有很好的文件支持。

1.2 操作系统的演变

为了理解 OS 的关键需求和 OS 主要特征的重要性, 回顾一下 OS 的发展是十分有益的。

1.2.1 串行处理系统

最早的计算机, 从 20 世纪 40 年代末到 50 年代中期, 程序员直接与硬件接触, 根本没有 OS。计算机运行在一个集成了指示器、各种开关、一些输入设备以及一个打印机的控制台之上。用机器代码编写的程序由输入设备、读卡机载入, 在因错误而导致程序被挂起时, 出错位置由指示灯显示。程序员可以通过检测寄存器和主存来寻找出错原因。如果程序正常执行完毕, 则结果会输出到打印机上。

早期的这种系统存在两个问题:

(1) 上机安排

上机请求要用一个订单来预约机器时间。典型问题是, 如果一个用户事先约定了一个小时, 而最终执行程序时只用了 45min, 就浪费了 15min 的时间; 如果在约定的一个小时内完成不了, 则程序就会在执行一小时后被迫停止。

(2) 启动时间

启动时间包括向存储器中装载编辑器和高级语言源程序、存储编译过的程序(目标程序)、然后连接装配目标程序与公共函数等所花的时间。其中的每一步都有可能引起纸带机或卡片机的安装和拆卸。如果这一过程出现错误, 则用户又不得不回到准备程序的开始。这样, 就有相当的时间被浪费在程序启动上。

OS 的这种工作模式被定义为串行处理模式, 它表明了用户串行获取计算机的事实。随着时间的推移, 各种系统软件工具应运而生, 包括标准函数库、链接器、装配器、反编辑器、I/O 驱动程序等, 它们对所有用户都是开放的。显然, 这些软件工具可以提高串行处理的效率。

1.2.2 简单批处理系统

早期计算机十分昂贵, 因而最大限度地利用它就显得很重要。早期计算机的上机安排和启动时间所造成的时间花费是不可接受的。