

S. E. 马德尼克 著
J. J. 多诺万

操作系统

科学出版社

操作系统

S. E. 马德尼克 J. J. 多诺万 著

石伍锁等 译

科学出版社

1980

内 容 简 介

本书从资源管理、进程和分层三个主要方面系统地论述了操作系统的功能、结构及其实现方法,还介绍了一个简单的示范性的操作系统实例。

全书共分九章:第一章为概论;第二章介绍了计算机硬件和程序设计方面的一些基本知识;第三、四、五、六章从资源管理的角度叙述了存贮、处理机、设备和信息几种不同资源的各种管理技巧,并通过实例说明各种管理技巧的使用情况;第七章给出了一个操作系统实例;第八章介绍了各种不同资源之间的相互制约关系,并介绍了估价性能的方法;第九章对几个具有代表性的操作系统,如 CTSS、MULTICS 和 VM/370 等,进行了剖析;最后为参考文献。

本书可供从事电子计算机专业的科技人员使用,也可供有关专业的大专院校师生参考。

Stuart E. Madnick John J. Donovan

OPERATING SYSTEMS

McGraw-Hill, 1974

操 作 系 统

S. E. 马德尼克 J. J. 多诺万 著

石伍锁等 译

*

科 学 出 版 社 出 版

北京朝阳门内大街 137 号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1980 年 11 月第 一 版 开本: 787×1092 1/32

1980 年 11 月第一次印刷 印张: 22 1/8

印数: 0001—6,520 字数: 507,000

统一书号: 15031·290

本社书号: 1802·15—8

定 价: 3.40 元

译 者 的 话

为了推动我国电子计算机技术的发展,以适应社会主义建设和四个现代化的迫切需要,我们翻译了 S. E. Madnick 和 J. J. Donovan 所著的《操作系统》一书,供从事电子计算机专业的同志们参考。

本书是根据美国麻省理工学院多年使用的电子计算机软件方面的教材而编写的。已被美国多所大学和研究机关作为教科书和参考书。作者从资源管理、进程和分层三个主要方面系统地论述了操作系统的功能、结构及其实现方法,还介绍了一个简单的示范性的操作系统实例。在范例中,不仅给出了系统的结构,描述了诸数据基和各例行程序的流程,还列出了该系统的全部汇编代码,这对于学习操作系统的的设计极为有益。

本书内容丰富、概念清晰、通俗易懂、理论联系实际,是一本较好的教学和专业参考书。它对于我国广大计算机工作者、大专院校计算机专业的师生以及其它与计算机有关的人员,均有较大的参考价值。

全书原为十章。第一章为全书的轮廓,介绍了操作系统的重要性、概念和术语、操作系统的发展历史以及研究它的几种观点。

第二章介绍了计算机硬件和程序设计方面的一些基本知识,包括机器结构、机器语言程序设计、I/O 程序设计和中断处理等内容。

第三、四、五、六这四章从资源管理的角度阐述了存贮、处

理机、设备和信息这几种不同资源的各种管理技巧，并通过实际例子说明各种管理技巧的使用情况。这几章的目的在于使读者能领会并掌握分析、设计操作系统的技术。

第七章是用前几章介绍的概念和技巧设计的一个教学用的操作系统范例。这个系统采用了目前广泛流行的进程和分层等概念，虽很简单，却包括了现代操作系统最基本的内容，读者可从中学到设计思想和方法。

第八章介绍各种不同资源之间的相互制约关系，并介绍了估价性能的方法。

第九章是用本书介绍的观点和技术来剖析几个具有代表性的操作系统，如 CTSS, MULTICS 和 VM/370 等。

第十章是带注释的参考书目，翻译时已将原有的注释略去，保留了原文书目，去掉了章号，改称参考文献。

原书还有附录 A 和 B，附录 B 是微程序设计简介，内容很简单，所以将其略去。另外，附录 A 中的部分内容也已被略去。

参加本书译校工作的有：王世荣、石伍锁、沈林兴、李赣生、陈崇连、张兴耆、贾跃良等。由于译者的水平有限，错误和不妥之处在所难免，恳请读者批评指正。

目 录

第一章 本书概貌.....	1
1-1 操作系统的重要性	2
1-2 基本概念和术语	3
1-3 操作系统——资源管理程序	9
1-4 操作系统——进程观点(这些资源管理程序在系统中进行活动).....	13
1-5 操作系统——分层观点和扩充机器观点	17
1-6 运用我们的观点举一个例子——OS/MVT	23
1-7 从其他观点看操作系统	26
1-8 一般的设计考虑	31
1-9 后面各章之序	33
1-10 小结.....	33
第二章 I/O 程序设计、中断程序设计和机器结构	40
2-1 机器结构	41
2-2 汇编语言程序设计	43
2-3 入出程序设计	59
2-4 中断结构和处理	76
2-5 I/O 和中断处理程序举例	89
2-6 小结	107
第三章 存贮管理.....	120
3-1 单一连续分配	122
3-2 多道程序介绍	125
3-3 分割分配	129
3-4 可重定位分割存贮管理	143
3-5 分页存贮管理	150

3-6	请求分页存贮管理	162
3-7	分段存贮管理	193
3-8	分段并请求分页存贮管理	212
3-9	其他存贮管理方法	216
3-10	存贮管理的发展趋势	219
3-11	小结	230
第四章	处理机管理	241
4-1	状态图	242
4-2	作业调度	248
4-3	进程调度	272
4-4	多处理机系统	283
4-5	进程同步	287
4-6	作业和进程结合起来调度	310
4-7	小结	311
第五章	设备管理	327
5-1	设备管理技术	328
5-2	设备特性——硬件考虑	330
5-3	通道与控制器	344
5-4	设备分配的考虑	348
5-5	I/O 交通管制程序, I/O 调度程序, I/O 设备处理程序	349
5-6	虚拟设备	357
5-7	设备管理的发展趋势	377
5-8	小结	377
第六章	信息管理	388
6-1	引言	388
6-2	一个简单的文件系统	391
6-3	文件系统的一般模型	394
6-4	符号文件系统	405
6-5	基本文件系统	409

6-6	存取控制验证	410
6-7	逻辑文件系统	414
6-8	物理文件系统	421
6-9	分配策略模块	426
6-10	设备策略模块, I/O 启动程序, 设备处理程序.....	429
6-11	信息管理的趋向.....	430
第七章	一个操作系统范例的设计.....	438
7-1	引言	438
7-2	系统概述	439
7-3	设计概观	441
7-4	范例操作系统的级和层	443
7-5	核心数据基和例程	448
7-6	处理机管理(低级模块)数据基	451
7-7	处理机管理(低级模块)例程	459
7-8	存贮管理的数据基	462
7-9	存贮管理例程	465
7-10	处理机管理(高级模块)数据基.....	467
7-11	处理机管理(高级模块)例程.....	468
7-12	设备管理数据基.....	473
7-13	设备管理例程和进程.....	474
7-14	作业管理例程和进程.....	477
7-15	用户程序和进程.....	484
7-16	SVC 流程部分踪迹.....	484
7-17	操作系统范例的程序表.....	489
第八章	相互关联问题: 性能估价.....	547
8-1	存贮管理	548
8-2	处理机管理	549
8-3	设备管理	551
8-4	信息管理	554
8-5	用户具有支配权情况下的折衷办法	555

8-6	影响	555
8-7	交换与分页的比较	558
8-8	抖动	564
8-9	小结	575
第九章 实例研究		579
9-1	引言	579
9-2	IBM 系统/360 和系统/370 的操作系统	580
9-3	兼容分时系统(CTSS)	594
9-4	多路信息计算系统(MULTICS)	609
9-5	虚机器/370 (VM/370)	628
参考文献		653
附录		667

第一章 本书概貌

本书的主要目的有两个：一是讲授高等软件工程基础（包括异步操作、中断处理、操作系统界面和硬软件权衡），一是使读者能够设计和分析操作系统。

现代计算机硬件功能很强（见图 1-1a），用途也很多。然而，硬件向外界提供的界面，却往往令人感到不便或难以使用，为了驯服这种“裸机”，我们研制了操作系统。操作系统能够管理基本的硬件资源，从而能为用户及其程序提供服务周到的界面（见图 1-1b）。操作系统对于高效率的计算机操作是必不可少的，因此许多人认为操作系统与硬件是不可分割的。

研究操作系统的重要理由很多。最主要的是：（1）为专门的用途，人们不得不设计自己的操作系统，或者对现有的系统进行修改；（2）大多数计算部门主要的决策是选择操作系统以及挑选该系统中的选择部分；（3）用户为了完成自己的任务必须与操作系统交往，因为操作系统是用户与计算机间的主要界面；（4）操作系统中所用的许多概念和技巧一般可以应用到其它领域。本书所介绍的轮廓对于阐述这些论点是特别适用的。

“操作系统”这一术语表示计算机系统内负责控制和管理处理机、主存、辅存、I/O（输入/输出）设备和文件等资源的一些程序模块。这些模块用于解决资源间的矛盾，千方百计优化性能并简化系统的有效使用，在用户程序与计算机硬件之间起接口的作用。

有时，人们把这些模块总称为控制程序、监督程序、执行

1108363

· 1 ·

程序、管理程序或操作系统。而面向应用的模块,如语言处理程序、库子程序、调试程序等,这里不加讨论。按我们给操作系统下的定义,这些程序不过是操作系统的用户而已,将在别处另做适当说明 (Donova, 1972; Knuth, 1968)。

1-1 操作系统的重要性

在一些应用方面,计算机对于社会早已是极其重要的了,如工资分发和银行业务,它们所需的计算量之大,已使计算机成为不可缺少的为其服务的工具。例如,在五十年代中期,“美国银行”就已经开创了计算机在银行界的应用,因为他们认识到,要是还用人工来处理支票,那末用不了多久,就需要整个加利福尼亚州的成人都来处理才行(见 Fano, 1972)。

目前,计算机已经在一些新的领域中使用,例如汽车设计、先进的医疗护理、空间开发等,现在还正在研究把计算机应用于学习、获取知识和人工智能等领域。

在计算机产品和计算机服务方面,每年要花费 200 多亿美元。据估计,其中有百分之七十用于软件的研制及维护、数据处理人员的工资以及其它与软件有关的项目上。各项费用的指标都表明,计算机的应用和可用性还在继续扩大。

靠什么来控制计算机?在用户和计算机之间起媒介作用的是什么呢?正是操作系统。操作系统是这样的一个程序:当它执行时,它能控制计算机的操作。例如,操作系统对计算机的使用进行调度。较完善的操作系统提高了机器的效率,从而降低了使用计算机的费用。例如,操作系统允许在一个用户打印的时候让另一个用户运行它的程序,从而提供了同时操作的功能。

本书介绍了操作系统的一个轮廓,用以研究、设计及编制操作系统。这个轮廓既包括小型系统(例如第7章操作系统范例),又包括大型系统(例如第9章实例研究中所介绍的OS/VS等)。

1-2 基本概念和术语

本节介绍计算机硬件结构、程序设计及操作系统诸方面术语的基本概念。

1-2.1 计算机硬件结构的术语

计算机硬件的配置有很多种,图1-1a画出了其中可能的一种。

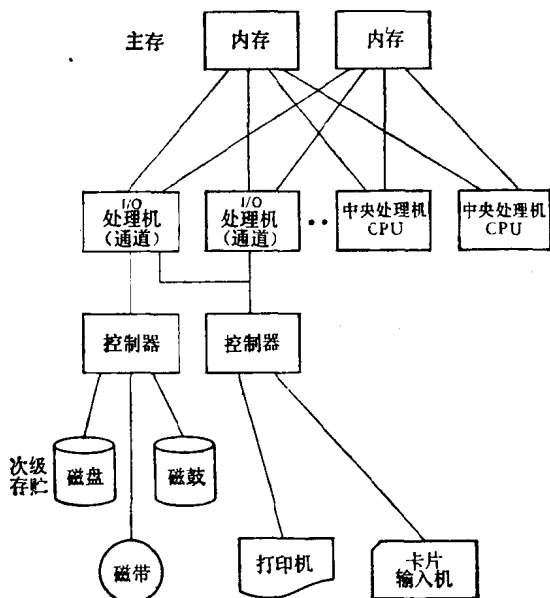


图 1-1a 基本的计算机硬件(裸机)

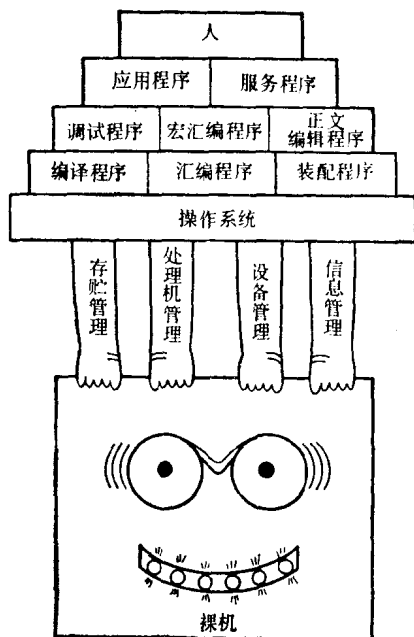


图 1-1b 操作系统与基本的计算机硬件之间的关系

指令和数据存放在主存中。主存以往都称为磁芯存储器（主存、内存、磁芯等术语往往可以替换使用）。与主存相连的是各种处理机。处理机是一种能够解释指令并执行指定操作（如加法）的硬设备。指令和数据是以编码的形式存于主存的。

一个计算机系统有一台或多台中央处理机（CPU）。CPU 能对内存中的数据进行加工及执行算术运算，也能执行一些控制其它处理机的指令。例如，CPU 可以执行一条启动 I/O（SIO）指令去启动一台 I/O 处理机。

在许多现代系统中，有些处理机被专门用来高效率地和（或）便宜地执行特定的任务。例如，I/O 处理机就是旨在用来控制 I/O 设备，处理主存与 I/O 设备间的数据传送的，这

种处理机。从简单的到相当复杂的都有。简单的 I/O 处理机只限于控制某一台 I/O 设备；而复杂的 I/O 处理机可以同时执行几条 I/O 指令并控制几台 I/O 设备。

几乎任何想得到的设备，从激光到挤奶机，都可以用作 I/O 设备。许多较常用的设备（卡片输入机、打印机、磁盘、磁鼓、磁带机等）均需要有公共的控制线路，以便能用于多台设备。由于经济上的原因，有时把这种公共的硬件独立出来，称之为控制器。

可能有许多用户程序竞争系统资源（例如主存、CPU、I/O 处理机、控制器、设备等）。哪一些用户将占有资源，可以占有多长时间？这些问题由什么来决定呢？当一台设备发生故障时，靠什么来防止系统免于崩溃呢？由谁来指定一条经过 I/O 处理机和控制器到达某一台设备的通路呢？谁来管理所有这些资源呢？所有这些问题的答案，都是操作系统。

1-2.2 程序设计术语

图 1-1a 表示一个计算机系统的硬件。大多数计算部门在软件方面的投资比硬件方面的投资还要大。软件是指用来执行某些任务的程序或数据的集合。程序是指令的序列（有时不大严格地把程序称为代码）。程序放在主存中，由处理机解释执行。程序不用的时候，往往放在某种辅存设备内，例如磁盘、磁鼓、或磁带上。

系统可以包含一些预先组装好的程序供用户使用（如开方、检索、分类等子程序、编译程序、汇编程序、翻译程序等）。对这些预先组装好的程序怎样管理呢？由谁来记录这些程序所存放的位置呢？还是操作系统。

一个孤独无助的用户，面对图 1-1a 那样令人生畏的系统，可能会感到沮丧。假定该用户是哈佛大学的学生，刚刚学

会编制 CPU 程序。他想把从穿孔卡片上读入的数据加上 49。他怎样把这些卡片读入计算机呢？他还必须学会编制 I/O 处理机的程序吗？很明显，这对于一个初学者来说要求太高了。幸好，操作系统一般都提供 I/O 处理机的程序。总之，操作系统能帮助用户来构造并执行他的程序。

操作系统提供了一个环境，以便使具体的软件程序组能够在这个环境中执行。例如编译程序，它把用高级语言写的源程序（如 FORTRAN 程序）翻译成相应的机器语言程序（即目标程序）。现代的编译程序与操作系统是密切合作的，不论是编译源程序时还是执行目标代码时都是这样。在编译期间，编译程序需要靠操作系统读入源程序，印出清单，并把它输出穿孔成卡片叠。像 PL/I 这样的编译程序所产生的目标程序，在执行时还需要一个很复杂的运行环境。ALLOCATE（分配）和 FREE（释放）语句需要求助于操作系统的存贮管理功能。PL/I 的 ON CONDITION 语句需要用到操作系统的中断处理程序。

1-2.3 操作系统术语

用户把作业提交给操作系统。用户是想要让计算机系统为他工作的人。作业是这种工作所需的活动的集合。一个作业可以分几步（作业步）。作业步是必须顺序进行的工作单位。例如，一个作业分三步：编译、装配和执行。一旦操作系统接受了某一用户的作业，它就可以创建多个进程。进程（或任务）是彼此可以并行运行的计算部分。图 1-2 表示用户、作业、进程和地址空间之间的相互关系。

对进程的另一种观点是：进程是处理机执行一组程序的运动轨迹。进程所存取的全部程序和数据组成地址空间。图 1-2 表示两个典型的地址空间——一个是 I/O 进程的地址空

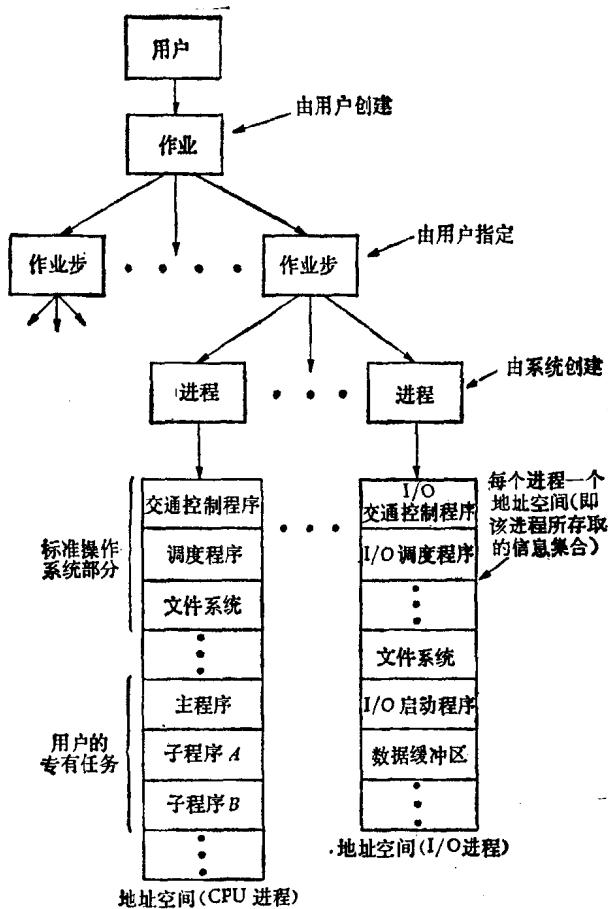


图 1-2 用户、作业、作业步、进程和地址空间

间,一个是 CPU 进程的地址空间。

注意,操作系统的文件系统部分的实际代码对这两个地址空间可以是相同的。如果代码以这种方式被共享,则它必须是纯的,否则必须用置锁的办法来避免竞争情况。纯代码(有时也称为再入代码)就是并不修改自身的程序。

操作系统必须把进程的地址空间转换成物理存贮空间。这件事可以由专门的硬件(如分页系统)来协助解决,也可以主要由软件来实现(如交换系统)。

如果一个系统可以使几个进程同时处于执行状态,那末我们就称该系统是多道程序设计的。进程处于执行状态的含意是:它的计算已经开始,但还未完成或未中止(错误结束)。(注意:一个进程可以处于执行状态而并不正在执行。即,已经算出了一些中间结果,但当前处理机并不正在此进程上运行)。

大多数现代计算机系统还有一个特征:它具有特殊的硬件和特殊的指令供操作系统使用。这些指令通常称为特权指令,一般的用户是不能使用的。大多数现代计算机至少有两种执行状态——目态(也叫用户状态、从态)和管态(也叫执行状态、主态)。管态时处理机能够正确地执行特权指令。很明显,要有一条特权指令用来改变机器的状态。还可有其它的特权指令用以启动 I/O 处理机,改变部分存贮的保护权,改变机器的中断状态等。

通常,可利用硬件保护来控制对部分存贮的存取。例如,可以规定操作系统的某些存贮部分是不能写入的,也就是说,禁止用户程序去修改操作系统程序。

中断机构硬件使操作系统能够协调几个同时进行的操作。它还可实现程序的非顺序执行流。中断是一种强使处理机关注某个事件的机构(类似于你在看书时可以被亲戚朋友所打断)。硬件可以屏蔽某些中断。

这里我们应注意,许多计算机装置中并不是都包含所有这些特殊指令或特殊硬件。(IBM 1130 就没有通常的屏蔽功能。)缺少这些功能,对于这样一种系统的用户或买主会有什么不便呢?另一方面,许多大型的系统却具有比上面所述的