高等学校教材

操作系统教程
—UNIX系统V实例分析

孟庆昌

西安电子科技大学出版社

高等学校教材

操作系统教程

——UNIX系统V实例分析

孟庆昌

西安电子科技大学出版社

1989

内 容 简 介

本书以 UNIX 系统 V 操作系统为实例，介绍计算机系统中的核心软件——操作系统。全书共分十二章。第一章概述操作系统的概念、发展历史、分类、功能及 UNIX 系统的特点。第二章至第六章分别介绍进程管理、处理机管理、存储管理、设备管理和文件系统。第七章介绍死锁概念。第八章至第十二章分别介绍并发程序设计、多处理机系统、分布式系统和操作系统的设计原则、性能评价及安全性。附录中给出了 UNIX 系统 V 的系统调用及常用命令。

本书可作为计算机科学和工程类专业的教材以及有关科技人员的参考书。

高等学校教材

操作系统教程

——UNIX 系统 V 实例分析

孟庆昌

责任编辑　李纪澄

西安电子科技大学出版社出版

西安电子科技大学印刷厂印刷

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 16 8/16　字数 387 千字

1989年 12 月第 1 版　1989年 12 月第 1 次印刷　印数 1 — 3 000

ISBN 7 — 5606 — 0095 — 6/TP · 0032　　定价: 3.30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、中等专业学校工科电子类专业教材的编审、出版的组织工作．由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978年至1985年已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用．

为了使工科电子类专业教材能更好地适应"三个面向"的需要，贯彻"努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材"的精神，我部所属的七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了1986～1990年的"七五"(第三轮)教材编审出版规划．列入规划的教材、实验教材、教学参考书等近400种选题．这批教材的评选推荐和编写工作由各编委会直接组织进行．

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会(小组)评选择优产生出来的．广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力．

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处．希望使用教材的单位，广大教师和同学积极提出批评建议，共同为不断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

操作系统是计算机系统的基本组成部分，它在整个计算机系统软件中占据核心地位。对操作系统的概念、理论和方法的研究以及对操作系统的使用、分析、开发和设计，历来是计算机领域中最主要的课题和任务之一。因而，操作系统是计算机科学教育的基本课程之一。它涉及到对各种资源(包括硬件和软件资源)的有效管理，又为高层软件的运行提供良好的工作环境，起到承上启下、纵横贯通的作用。

本教材是在前几年教学实践的基础上，吸收国内外较新的理论和技术，参照原教育部"计算机软件专业操作系统教学大纲"和美国 IEEE(The Institute of Electrical and Electronic Engineers)1983 年推荐的教学大纲中对核心课程"Operating Systems"的要求进行编写的。

本书注意在理论上力求系统、完整，尽量体现当代的研究成果，抛弃一些陈旧的概念和说法。在讲授方法上，注意理论与实际的结合，特别是以当代最流行的 UNIX 操作系统为实例，介绍了一个操作系统内部结构的全貌和某些实现技巧。在内容安排上，注意由浅入深，由一般到具体，先介绍操作系统的概念和服务功能，然后一一讲述这些功能的实现和算法分析，最后介绍有关操作系统设计的理论和最新技术。

本书共分十二章。第一章概述操作系统的发展历史、分类、功能、体系结构及 UNIX 系统的特点。第二章介绍进程的基本概念、有关进程的操作、进程通信和中断处理。第三章介绍处理机管理，包括三级调度功能及常用算法。第四章到第六章分别介绍存储管理、设备管理、文件系统的功能、概念及其主要实现技术。第七章介绍死锁的概念、死锁的产生及解决办法。第八章介绍并发程序设计。第九章到第十二章分别介绍多处理机系统、分布式系统、操作系统的设计原则和性能评价及安全性等。在附录中给出了 UNIX 系统Ⅴ的系统调用及常用命令。

每章的开头，先交待所要讨论问题的由来、环境和意义，然后一层层逐渐展开论述，在讲授理论的基础上，辅以 UNIX 系统的实例，从而加深对概念的理解和形象化思维，最后对本章内容进行小结，对主要概念和理论进行汇总。每章后面附有大量习题。这些有代表性的习题对学习巩固正文中的知识是有益的。

本书以 UNIX 系统Ⅴ为实例进行讲述，既有理论价值，又有实用意义。目前 UNIX 系统已成为国际上公认的一种最流行的分时系统。它集中了以往操作系统的精华，本身有一系列特点，在 16 位、32 位微处理机或超级微型机、小型机的操作系统中占据主导地位。在美国、加拿大及西欧、日本等国家和地区，大多数高等院校中都把 UNIX 系统用于教学和科研领域。目前对这个系统的应用和开发方兴未艾。系统Ⅴ是 UNIX 操作系统的最新版本。

本书是计算机科学和工程类专业的教材，也可供有关科技人员参考。在条件允许的情况下，尽量让学生在 UNIX 系统环境下上机实习。书中带＊号的部分可酌情作选学处理。

本书的主审单位是上海交通大学。主审人是尤晋元副教授，他对本书的编写给予了积极的支持和帮助，并提出了很多建设性的意见。上海交通大学徐良贤副教授、中国科学院软件所孙玉方副研究员和北京信息工程学院张海藩副教授都对本书的编写给予了热情的支

持和帮助．北京信息工程学院科研处刘振英同志为编者做了大量的资料准备工作．在本书编写过程中，还得到过教材科钱萍同志及其他同事的帮助和支持．特在此一并表示诚挚的谢意．

由于编者水平有限，时间又很紧迫，没有来得及反复推敲和修订，因而书中一定还会有错误和不当之处，恳切期望广大读者给予指正．

编　者

于北京信息工程学院

1988 年 9 月

目　录

第一章　操作系统概述

1.1　计算机发展简史

从 1946 年世界上第一台电子数字计算机(以下简称计算机)问世以来，至今短短的 40 多年间，它的性能得到飞速的提高，其应用领域遍及社会生活的各个角落，相应地计算机技术和理论都得到迅猛发展。在当今信息时代，计算机技术已处于举足轻重的地位。

计算机的发展历史大致可分为以下几个阶段:

第一代: 1946 年～1959 年，以美国建造的 ENIAC 为代表，主要电子器件是电子管。

第二代: 1960 年～1964 年，主要特征是以晶体管为主要电子器件，如 IBM 7090 系列。

第三代: 1965 年～1973 年，以集成电路作为计算机的主要器件，如 IBM 360 机种。

第四代: 从 1974 年至今，大规模集成电路(LSI)、超大规模集成电路(VLSI)用于计算机，从巨型机到微型机、个人计算机，其类型层出不穷。与之相应，计算机科学和技术也得到了迅速发展。

随着各种高技术特别是人工智能技术在计算机上的综合应用，新一代的计算机不久将会出现。

1.2　操作系统的发展过程

第一代计算机初创时，相应的软件也处于初期发展阶段。这时，仅有汇编语言和一些标准子程序，对计算机的操作采用手工方式，操作系统还未形成。计算机发展到第二代，软件也得到了显著的发展，先后出现了象 FORTRAN、ALGOL 等高级程序设计语言，从而完成了编译程序。此外，又出现了成批处理、执行系统以及多道成批系统和分时系统，这标志操作系统已基本形成。计算机进入第三代，软件得到更大发展，出现了许多通用和专用的高级语言，操作系统开始普及，实时系统和通用操作系统也伴随出现。在这一阶段后期，出现了许多大型计算机，出现了数据库管理系统、远程成批处理和计算机网络。随之产生的问题是系统越来越复杂，研究、设计和实施这些系统不仅需要巨大的工作量，而且其正确性也往往难于保证，这迫使人们对已有的操作系统进行总结，对它的重要概念、功能设计、结构设计以及实现算法等方面重新加以分析和研究，从而在理论上和工程实践上取得很大成果。美国贝尔(Bell)实验室于 1970 年研制的 UNIX 操作系统(以下简称 UNIX)就是其成功代表。计算机进入第四代以后的 10 年左右，微型机迅速兴起，造价越来越低，而性能越来越优；各种巨型机的结构正经历划时代变革；各种软件系统得到飞速发展，越来越向标准化、智能化、工程化方向前进，出现了各种局部计算机网络系统、远程计算机网络系统、分布式系统、众多的专家系统和复杂的工作站等。

下面对操作系统的各个发展阶段作一简要介绍。

1.2.1 手工操作阶段

早期计算机是十分庞杂的由控制台"指挥"的机器．程序员编好程序，然后在操作员控制台上直接操作这个程序．首先，在控制台上把全机置成初态，利用纸带机、前面板开关或者卡片机，由人工把程序装入内存，然后启动执行它．在程序运行期间，通过控制台上的显示灯，程序员和操作员可以监视它的执行情况．如有错，则中止该程序，检查内存和寄存器的内容，直接由控制台对它进行调试．最后，打印结果，或者把输出在纸带或卡片上穿孔，以备以后打印．

这是初级人机交互方式．这种使用方法具有以下特点：资源独占，即计算机的全部硬件资源(如 CPU、内存、设备等)都由一个程序独自占用；串行工作，人的操作与计算机的运行以及计算机各个部件之间都是按时间先后顺次工作的；人工干预，计算机是在人的直接联机干预下进行工作的．上述控制关系如图 1-1 所示．

这种工作方式有严重的缺点：一个是资源浪费，另一个是使用不便．例如，你预先设定的运行程序要用 1 小时的计算机时间，但实际上在 1 小时内不能完成，到时也必须停止，下机，让后面的预约用户使用．你要继续做时，还得再约机时．另一方面，如你的程序提早运行完，那么在剩余的时间内计算机闲置，造成机时浪费．其实，由于手工操作的延迟以及挂起查错的缘故，大量的计算机时间被耗费掉了．

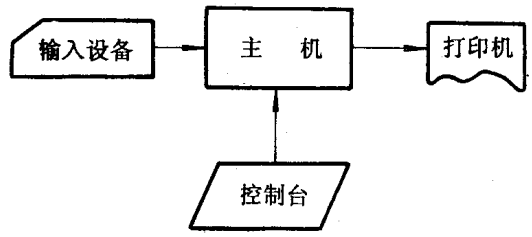

图 1-1　第一代计算机的控制关系

随着计算机运算速度的大幅度提高，各种部件和设备日益增多以及计算机应用范围的普及，上述缺点就愈加突出了．

1.2.2 早期批处理阶段

在计算机发展初期，其速度较慢，手工操作降低效率的矛盾尚不突出．例如，一个程序在速度为每秒一千次的计算机上运行，设需要 1 小时的时间，而人工进行其它操作(装入程序、启动运行等)所需的时间设为 5 分钟，则 CPU 有效运行时间与 CPU 空等时间的比例为 12∶1．若该程序在每秒 6 万次的计算机上运行，有效运行时间仅需 1 分钟，则两者时间的比变为 1∶5．计算机速度越高，这种机时浪费就越惊人．为解决这个问题，就必须缩短建立作业和人工操作的时间，人们首先提出了从一个作业转到下个作业的自动转换方式．从而出现了早期批处理方式．它又分为联机批处理和脱机批处理两种类型．完成这一自动转换工作的程序叫做监督程序．

1. 早期联机批处理

在这种系统中，操作员有选择地把若干作业合为一批，监督程序先把这批作业从输入设备上逐个地转输到磁带上，当输入完成，监督程序就开始执行这批作业．监督程序从磁带上顺次把每个作业读入到内存，对它的程序进行汇编或编译，然后由装配程序把编译后的目标程序及其所需的子程序(目标程序形式)一起连接成一个可执行的目标程序，再装入内存后立即启动执行它．计算完成之后，由善后处理程序输出该作业的计算结果．第一个

作业全部完成之后，监督程序又自动调入该批的第二个作业，并重复以上过程，直至该批作业全部处理完毕，再把下一批作业输入到磁带上，并按上述步骤处理。如此继续，就实现了作业间的自动转换，从而缩短了建立作业和人工操作的时间。

在这种联机批处理方式下，作业信息从卡片传送到磁带上，再从磁带调入内存，以及计算结果在打印机上输出，统统由中央处理机(简称 CPU)直接控制完成。这样在输入或输出过程中，主机速度得不到发挥，降为慢速外设的水平。随着计算机速度的提高，慢速外设与高速主机间串行工作的矛盾就更加突出了。为克服这一缺点，人们在成批处理中引进了脱机输入输出技术。

2. 早期脱机批处理

这种方式的明显特征是在主机之外另设一台小型卫星机，该机只与外部设备打交道，不与主机直接连接，从而使主机腾出较多的时间专门完成快速的计算任务。其结构模型如图 1-2 所示。

它的工作过程是：读卡机上的作业通过卫星机逐个地传送到输入磁带上，而主机只负责把作业从磁带上调入内存并运行它，作业完成后，主机把计算结果和记帐信息记录到输出带上，由卫星机负责把输出磁带上的信息读出来，并交打印机印出。这样，就使主机与慢速外设由以前的串行工作变成了并行工作，而且主机是与较快速的磁带机进行信息交换，从而提高了主机效率。由于卫星机仅处理简单的输入、输出工作，因而只须采用较小型的机器即可。

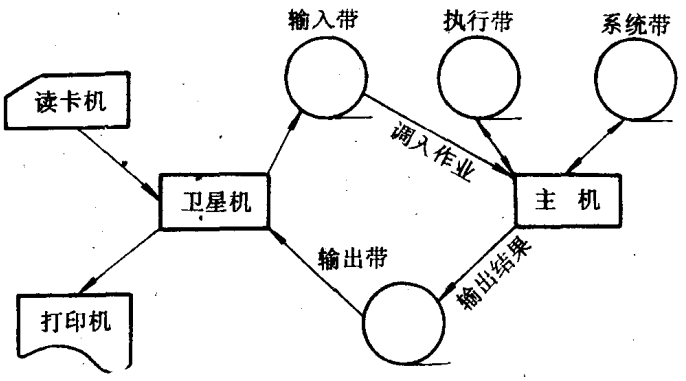

图 1-2 早期脱机批处理模型

早期批处理系统提高了资源的利用率，同时推动了其他软件的发展，出现了标准输入输出程序、汇编程序、FORTRAN 编译程序、装配程序和库例行子程序等，也产生了连接装配程序、程序库以及程序覆盖等新的程序设计技术。解题操作过程变成"装入／解释(或编译)／装配／执行"等四个步骤。从而使上机操作初步自动化，程序员可以不进入机房，而由操作员完成必要的手工操作，方便了用户使用。

1.2.3 执行系统阶段

虽然批处理系统具有减少人工干预、提高主机效率的优点，但又带来必须解决的保护问题。因为在上述过程中，所涉及到的监督程序、系统处理程序和用户程序之间是一种相互调用关系，并无主次之分。程序的控制转移是利用一定的调用方法实现的，这样就存在潜在的危机。一旦目标程序中出现非法指令或用户程序陷入无穷循环("死循环")，整个系统就无法正常工作而处于停滞不前状态。更严重的是无法防止用户程序破坏监督程序和系统处理程序，为解决这个问题引进了执行系统。

60 年代初期，引入通道，它是一种硬件机构，独立于CPU而直接控制一台或多台外部

设备与内存之间进行数据传送．实际上，通道就是一台功能单一、结构较简单的处理机．一旦主机发出启动外设的指令并提供给通道必要的参数信息(例如，数据在内存的地址；外设、通道及控制器的编号或地址等)后，通道就独立地完成交给它的I／O工作．主机与通道同步的办法，最初是借助一条询问指令实现的．主机发出询问指令，询问通道工作完成否．如未完成，主机就反复地询问等待，直至通道工作结束．

为了使通道传输与主机运行充分并行，以减少主机的询问等待时间，引进了缓冲技术．主机所需的输入信息由通道提前输入到缓冲区，同样，主机的输出信息先送到输出缓冲区，再由通道输出，从而减少主机等待时间．但是这些方法未能彻底解决问题，于是引入了中断机构．

最初，中断是指外部设备或通道向主机报告情况的一种通信方式．当外设或通道完成I／O工作或传输过程中发生错误及故障时，便向主机发送一个信号使之停止正在执行的工作，而转去执行为处理该信号而设置的程序，中断处理完毕后主机再回到原来的工作点继续执行．这种中断一般称为I／O中断或设备中断．借助中断这一手段较好地解决了主机与通道的同步问题，主机不必动态地询问I／O完成否，而是在启动通道后继续执行程序，直至被通道发来的中断信号所打断．这样，主机与通道就最大限度地保持并行性．

受I／O中断的启发，人们又引进了其它中断，如程序中断(算术溢出或非法指令等)、时钟中断等，从而克服了以往的出错停机、程序死循环的毛病．

通道和中断机构的引进使外部设备的管理更加复杂，因而在系统中增加了中断处理程序和输入输出控制程序(IOCS)，对所有程序都起着指挥和控制的作用．因此一般让它们常驻内存，而让另外一些系统处理程序放在外存中以供调用．这些常驻内存的程序称为执行程序(有时也称控制程序、监督程序、管理程序等)．执行程序与前述监督程序有明显差别，它与其它程序的关系不是相互调用关系，而是一种控制关系．其它程序在它的指挥控制下工作，从而增强了保护能力．

虽然在这一阶段还未出现对内存的硬件保护手段，但执行系统的出现促进了软件技术的发展，比较主要的有两个方面：一是系统程序模块化，即把系统的每一个相对独立的功能孤立在某一程序中，以后对它进行调整和修改时，不涉及其它模块；各模块之间通过固定通信区或指定单元实现接口联系，从而使系统结构简单、易于修改和扩充．二是开始出现了作业控制语言(JCL)．不过这一时期的JCL语言主要是控制卡式的，即把所需的命令穿成卡片插在作业卡片叠中，并在其第一列穿上特殊字符(如"$")，以便标志它是控制卡．(实际上在批处理阶段已出现这种作业控制方式，只是当时的作业控制过于简单．)

1.2.4 多道程序系统阶段

不论是早期批处理还是执行系统，都是单道顺序地处理作业．这种一个作业处理完才处理另一个作业的串行办法，仍然妨碍系统效率的充分提高．例如，处理以计算为主的作业时，CPU很忙，但外部设备却空闲；而对那种以I／O为主的作业(计算量很少)，又会造成CPU的空载．为了克服这些缺点，促进了多道程序系统的出现．

多道程序设计的基本思想是在内存里同时存放若干道程序，它们可以并行地运行，也可以交替地运行．这样处理机得到了比较充分的利用．例如，当某一道程序因为某些原因(如需要安装磁带、等待键盘打入的命令或输入输出操作完成)而不能继续运行时，就把

CPU 转给另外的作业并执行它，当后者也需要等待时，CPU 又转给其它作业，等等．以后，当条件满足第一个作业运行时，又可重新分到 CPU 而继续运行下去．只要存在某些可执行的作业，CPU 始终不会空载．图 1–3 表示了一个具有两道程序的系统中 CPU 和通道的利用情况．

由图可见，在单 CPU 的系统中，这些程序在微观上只能是交替地运行，但在宏观上(在一段较长时间内)它们可被视为是并行的，因为在这段时间内各个可执行的程序都向前推进了．只有在多处理机系统中，这些并发程序才可以真正并行地执行．

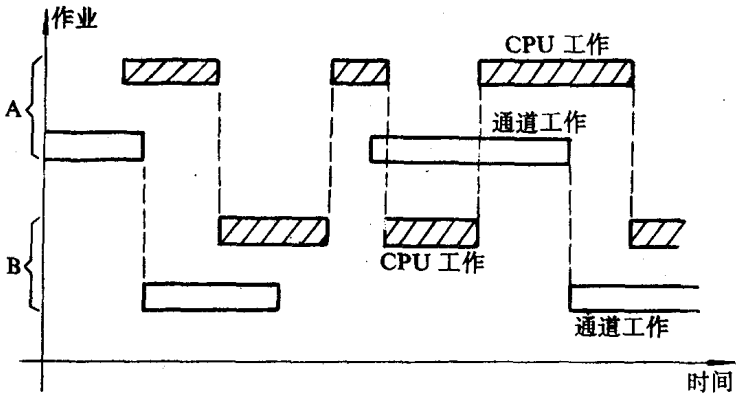

图 1–3　多道程序执行过程示意

在通道、中断机构及多道程序技术引入之后，人们又把输入、输出的任务交由主机去做，这样通道只实现内存与外设之间的数据传送，而其它处理工作则由常驻内存的若干系统作业来完成．这种技术就是 SPOOLing 技术(或称假脱机技术)．

在多道程序系统中，并发程序要共享系统内资源，使系统管理变得很复杂，从而对软件和硬件都提出许多新课题，也促进了它们的进一步发展，解决了系统保护、存储分配和简单的动态地址翻译等问题．例如，为了对操作系统程序(特别是其核心部分)进行保护，防止用户程序的破坏，提供了几种处理机状态——如“管理态”(简称“管态”)和“用户态”(简称“目态”)．当操作系统程序执行时，处理机处于管态，这时可以执行特权指令(一般用户不能使用它们)；而用户程序则在目态下执行．用户程序想得到操作系统服务只能通过执行硬件提供的特定指令，例如访管指令(IBM 370 的 SVC 指令，PDP–11 的 trap 指令等)，经严格检查才能进入管态下的操作系统程序中．

随着计算机的发展，计算机硬件和操作系统的关系会越来越密切，硬件机构进一步推动操作系统的发展，而操作系统也向硬件提出更新的要求．

1.3　什么是操作系统

操作系统是计算机用户和计算机硬件之间的接口程序模块，它是计算机系统的核心控制软件，其职能是控制和管理系统内各种资源，有效地组织多道程序的运行，从而为用户提供良好的工作环境，达到使用方便、资源分配合理、安全可靠等目的．

设置操作系统的第一个目的是扩充机器功能以方便用户使用．计算机系统的基本资源包括硬件(如处理机、内存、各种设备等)、软件(系统软件和应用软件)和数据．操作系统要保证整个系统在运行时各个用户对这些资源的需求，提供一台比裸机功能更强、且易于用户使用的机器．也就是说，用户面前所使用的计算机是经操作系统进行功能扩充之后的

虚拟机，其特性不同于作为其运行基础的物理机器．设想一下：如果你所用的机器上没有装入操作系统，那么你无法使用命令，不能利用应用程序，连设备、内存等都需要你亲自管理，那将是一种多么"可怕"的局面呀！从而也看出操作系统的重要存在价值．

设置它的第二个目的就是提高计算机系统的工作效率．这对多用户共享的大型系统来说特别重要．这些系统的造价昂贵，所以更应尽可能提高效率．操作系统本来就是资源管理程序，由它管理计算机系统各种资源，控制、协调各用户对这些资源的利用，尽可能地发挥资源的作用．作为"管理者"，操作系统主要负责如下事情：

- 监视各种资源并随时记录它们的状态；
- 实施某种策略以决定谁获得这种资源，何时获得，获得多少；
- 分配这种资源供需求者使用；
- 回收这种资源，以便再分配．

还可从其它角度看待操作系统，如把它视为控制程序，由它控制用户程序的执行，避免对计算机错误或不恰当的使用，特别是对I／O设备的控制和管理．

应该指出，对操作系统尚无严格的定义．通过操作系统做什么来定义它，或许比直接说它是什么更容易理解．

1.4 操作系统的服务功能

操作系统为程序运行提供一种环境，它为程序和用户提供某些服务．当然，各个操作系统所能提供的服务并不完全相同，但有些是共同的．操作系统的功能为程序员带来了方便，使程序编制工作变得容易了．操作系统的服务通常有以下几个方面：

① 程序执行．用户要执行程序，系统必须把它们装入内存，然后再运行．当程序正常完成或发生意外事故而无法运行下去时，必须把它终止．

② I／O操作．正在运行的程序往往需要进行输入和输出，包括文件读写和I／O驱动．专用设备需要专门的程序(如倒带驱动、CRT的清屏幕，等等)．由于用户程序并不能直接执行I／O操作，所以操作系统必须做这些事情．

③ 文件系统管理．用户的程序和数据需要建立文件才能保存在系统中，以后我们可以按照名字对它进行读写操作，当它无存在必要时可以按名字删除它，等等，这些复杂的工作都可由操作系统完成．

④ 出错检测．操作系统需要经常了解可能出现的错误．错误来源是多方面的，可能在CPU和内存硬件中出现(如内存出错或掉电)，也可发生在I／O设备上(如磁带机的奇偶错、读卡机的卡片干扰或打印机脱纸)，还可能由于用户程序有错(如算术溢出、地址异常或占用CPU时间过长)．操作系统对每类错误都要检测到，并采取相应措施，保证计算的一致性．

操作系统除上述为用户提供的服务外，操作系统还具有资源分配、统计、保护等另外一些功能，以保证它本身高效率地执行操作，从而使多个用户能共享系统资源，提高系统的效率．

⑤ 资源分配．多个用户或者多道作业同时运行时，每一个都必须分得相应的资源．系统中各类资源都由操作系统统一管理，象CPU调度、内存分配、文件存储等都有专门

的分配程序，而其它的资源(象I/O设备)有更为通用的申请和释放程序。

⑥ 统计。往往我们希望了解各个用户对系统资源的使用情况，如使用什么类型的资源，用了多少个等，以便用户付款或者简单地进行使用情况统计，作为进一步提高服务性能，对系统进行组合的有价值的工具。

⑦ 保护。在多用户计算机系统中文件主能对所创建的文件进行控制使用，并规定其他用户对它的存取权限。此外，当多个不相关作业同时执行时，一个作业不干扰另外一个作业。在多道程序运行环境中，对各种资源的需求经常发生冲突，为此，操作系统必须作调节和合理的调度。

操作系统的服务可以通过不同的方式提供，其中两种基本的服务方式就是系统调用和系统程序，它们各具特点。

1.4.1 系统调用

系统调用是运行的程序和操作系统之间的接口，通过它来处理更基层的服务。这些调用通常是作为汇编语言的指令来使用的(在UNIX系统上，系统调用可以用函数调用的形式出现)。系统调用可大致分为三个主要范畴：进程或作业控制、设备和文件管理以及信息维护。下面分别作简要介绍。

1. 进程或作业控制

简要地说，进程(process)是指程序的一次执行过程(详见第二章)。正在运行的进程可能需要正常(结束)或非正常(失败)终止执行。如果该进程在它的输入中发现了错误并要非正常终止，需要给它规定一个错误级别，严重的错误可以有较高的错误级别。我们可以把正常和非正常终止合在一起对待，把正常终止的错误级别定为0。

一个进程或作业在执行程序过程中可能要装入和执行另一个程序。例如，允许控制卡解释程序执行由用户作业控制卡所指示的程序。一个相关的问题是当被装入的程序终止后控制返回到什么地方。这个问题与下列因素有关：原来的程序是丢失了还是保留着，或者允许与新程序继续并发执行。

当新程序终止时，若控制返回到原来程序，我们必须保留原来程序的内存映象，并要有一套相应的机构。如果二者并发执行，我们必须创建一个新作业或进程。往往有一个专门的系统调用实现这个工作(创建进程或提交作业)。

如果创建了新作业或进程，甚至是一组作业或进程，我们能够控制它们的执行，这需要能对作业或进程的属性进行测试和重置，包括它的优先级、运行时间，等等(取进程属性和置进程属性)。当发现前面创建作业或进程不正确或无存在必要时，我们也要终止它。

创建新作业或进程后，往往需要等待它们完成工作。为此，可用系统调用来实现等待某段确定时间(等待时间)，但更多的是用系统调用来等待一个特定事件(等待事件)。当该事件发生(信号事件)时，这些作业或进程就发出信号通知等待者。

另一组系统调用在调试程序中很有用。很多系统都提供了转储(dump)内存的系统调用，这个预防措施对调试汇编语言或机器语言是有用的，尤其在批处理系统中。程序追踪(trace)列出了程序执行过程中用到的每条指令。

很多系统还提供了程序时间分布图，它表明该程序在特定位置或某个程序区间执行时的时间统计。时间分布图既需要追踪工具，也需要准确的时钟中断。