

高等学校
工科电子类规划教材精选系列

计算机操作系统

汤子瀛 哲凤屏 汤小丹



西安电子科技大学出版社

TP316

T38-3

401928

高等学校工科电子类规划教材精选系列

计算机操作系统

汤子瀛 哲凤屏 汤小丹



西安电子科技大学出版社

1996

(陕)新登字 010 号

JS146/29

内 容 简 介

本教材介绍计算机系统中的一个重要系统软件——操作系统(OS)。全书分四部分,共15章。第一部分介绍OS的基本原理,共10章,第1章概述了OS的形成、类型和功能等;第2~4章分别阐述了进程和线程的基本概念、同步和通信、调度和死锁;第5~6章为存储器管理和虚拟存储器;第7章为设备管理;第8、9两章分别介绍了文件系统和磁盘存储器管理;第10章为用户接口。第二部分中的第11和12章,分别介绍了网络OS和分布式OS。第三部分(第13、14章)介绍了一个OS实例——UNIX系统V的内部结构。最后一部分(第15章)介绍Windows的使用。

本教材可作为计算机和计算机应用专业的教科书,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。



高等学校工科电子类规划教材

精选系列

计算机操作系统

汤子瀛 哲凤屏 汤小丹

责任编辑 李纪澄

西安电子科技大学出版社出版

西安市长青印刷厂印刷

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 31 12/16 字数 758 千字

1996年12月第1版 1996年12月第1次印刷 印数 1-20 000

ISBN 7-5606-0496-X/TP·0232(课) 定价: 27.00 元

前 言

从 1981 年至今,本教材的第一作者一直负责电子部规划教材《计算机操作系统》主要编著和修订工作。这次我们的新版《计算机操作系统》又被西安电子科技大学出版社选人该社“工科电子类规划教材精选系列”,我们对西电出版社致以诚挚的谢意。

一本受读者欢迎的优秀教材,应该是作者在教材所涉及的领域中长期从事教学和科研的产物,是一个不断积累和改进的过程,必须不断地锤炼和突出教材的基本内容,又要不断地更新使之反映当代的新技术和新方法。我们就是本着这样的指导思想,从事本教材的编写工作的。

正因为教材的编写和改版是一个连续的过程,这次本教材的出版,也必然会联系到过去作者在这方面的教材编写工作。《操作系统原理》于 1981 年出版,1984 年对它进行了第一次重大修改,将 RTOS 实例换为 UNIX 系统,并改名为《计算机操作系统》,作为电子工业部规划教材出版。该教材曾获得 1987 年电子工业部优秀教材二等奖,在使用 8 年后,又于 1992 年对 1984 年版的教材做了较全面的修改,出版了《计算机操作系统(第二版)》,其中,增加了多处理机操作系统、网络和分布式操作系统两章,并用 UNIX 系统 V 取代了 UNIX V.6 版本。该教材获得 1996 年电子工业部优秀教材一等奖。

尽管 1992 年版的教材获得好评,但由于 OS 是计算机领域中最活跃的学科之一,其发展极为迅速,为使教材内容能及时地反映时代潮流,我们于 1996 年又对 1992 年版的教材进行了重写,调整了该教材的结构,由原来的 10 章改为 15 章。同时,也对教材的内容做了较全面的更新,其内容基本上能反映出 90 年代初期和中期的 OS 发展情况,即对 90 年代推出的 OS 中所采用的新方法和新技术大都在教材中有所反映。

新版《计算机操作系统》全书共分 15 章。第 1 章仍为 OS 引论,介绍了 OS 的形成、类型、功能和 OS/2 操作系统。第 2~4 章为进程管理,引入了能进一步提高程序并发执行程度的多线程概念,加强了对管理应用的介绍,增加了实时调度和多处理机调度的内容,并对 OS/2 的进程管理作了较详细的阐述。第 5、6 章分别介绍了存储器管理和虚拟存储器,增加了两级、多级页表和反置页表机制等内容,加强了对虚拟存储器性能的分析,并对 OS/2 操作系统,亦即对 Intel 80386

中的段页式存储管理方式做了扼要的阐述；第7章介绍设备管理。第8、9章分别介绍了文件系统和磁盘存储器的管理。其中，第8章着重阐述逻辑文件系统的结构和管理以及文件的保护；第9章则着重介绍了物理文件系统的结构和管理，并详细论述了能保证系统可靠性的数据一致性技术和磁盘容错技术，以及可提高文件系统性能的高速缓冲方法和磁盘调度方法。第11~14章介绍网络OS、分布式OS，以及对UNIX系统V内部结构的分析，它较之1992年版中的相应内容也有较多的更新。第10和15章分别介绍用户接口和OS的使用，主要是以Windows 3.2为背景，介绍图形用户接口和对Windows的使用。

本教材的第8、9章和第13、14章由哲凤屏编写，第10和15章由汤小丹编写，其余的九章均由汤子瀛编写。全书由汤子瀛统一编排定稿。

本教材在编写过程中，得到了西安电子科技大学出版社的大力支持与合作。此外，汤蓓莉、王佩雅等同志在整理、校对、绘图等工作中，都付出了艰辛的劳动，使本教材能如期地与读者见面。在此谨向以上各位表示衷心感谢。

本教材虽经多次反复修订，突出了操作系统的基本概念，反映了当代操作系统的新技术，但限于编者水平，在本次编写的教材中，仍难免会有错误和不当之处，恳请读者批评指正。

作者

1996年7月

目 录

| | | | |
|---------------------------|----|---------------------------------|----|
| 第1章 操作系统引论 | 1 | 2.2.4 进程控制块 PCB | 43 |
| 1.1 操作系统的目标、作用和模型 | 1 | 2.3 进程控制 | 46 |
| 1.1.1 操作系统的目标 | 1 | 2.3.1 操作系统内核 | 46 |
| 1.1.2 操作系统的作用 | 2 | 2.3.2 进程的创建 | 47 |
| 1.1.3 操作系统的层次模型 | 3 | 2.3.3 进程的终止 | 49 |
| 1.2 操作系统的发展过程 | 4 | 2.3.4 进程的阻塞与唤醒 | 50 |
| 1.2.1 推动操作系统发展的主要动力 | 4 | 2.3.5 进程的挂起与激活 | 51 |
| 1.2.2 无操作系统时的计算机系统 | 5 | 2.4 线程的基本概念 | 52 |
| 1.2.3 单道批处理系统 | 6 | 2.4.1 线程的引入 | 52 |
| 1.2.4 多道批处理系统 | 7 | 2.4.2 线程与进程的比较 | 53 |
| 1.2.5 分时系统 | 9 | 2.4.3 用户级线程和内核支持线程 | 54 |
| 1.2.6 实时系统 | 12 | 2.4.4 Solaris 操作系统中的线程 | 55 |
| 1.3 操作系统的特征和服务 | 13 | 2.5 OS/2 操作系统中进程的描述与控制 | 56 |
| 1.3.1 操作系统的特征 | 13 | 2.5.1 会晤的描述与控制 | 56 |
| 1.3.2 操作系统的服务 | 15 | 2.5.2 进程的描述与控制 | 58 |
| 1.4 操作系统的功能 | 17 | 2.5.3 线程的描述与控制 | 59 |
| 1.4.1 存储器管理的功能 | 17 | 习题 | 60 |
| 1.4.2 处理机管理的功能 | 18 | 第3章 进程的同步与通信 | 61 |
| 1.4.3 设备管理的功能 | 20 | 3.1 进程同步的基本概念 | 61 |
| 1.4.4 文件管理的功能 | 21 | 3.1.1 临界资源 | 62 |
| 1.4.5 用户接口 | 22 | 3.1.2 临界区 | 63 |
| 1.5 操作系统的进一步发展 | 24 | 3.1.3 利用软件方法解决 进程互斥问题 | 64 |
| 1.5.1 微机操作系统 | 24 | 3.1.4 利用硬件方法解决 进程互斥问题 | 67 |
| 1.5.2 多处理机操作系统 | 25 | 3.2 信号量机制 | 68 |
| 1.5.3 网络操作系统 | 26 | 3.2.1 整型信号量机制 | 69 |
| 1.5.4 分布式操作系统 | 29 | 3.2.2 记录型信号量机制 | 70 |
| 1.6 OS/2 操作系统简介 | 30 | 3.2.3 信号量集机制 | 71 |
| 1.6.1 OS/2 的引入 | 30 | 3.3 经典进程同步问题 | 73 |
| 1.6.2 OS/2 的主要性能 | 31 | 3.3.1 生产者—消费者问题 | 73 |
| 习题 | 34 | 3.3.2 读者—写者问题 | 75 |
| 第2章 进程的描述与控制 | 35 | 3.3.3 哲学家进餐问题 | 77 |
| 2.1 前趋图和程序执行 | 35 | 3.4 管程机制 | 78 |
| 2.1.1 前趋图的定义 | 35 | 3.4.1 管程的引入 | 79 |
| 2.1.2 程序顺序执行 | 36 | 3.4.2 管程的基本概念 | 79 |
| 2.1.3 程序并发执行 | 36 | 3.4.3 利用管程解决生产者 —消费者问题 | 81 |
| 2.1.4 程序并发执行的条件 | 38 | 3.4.4 利用管程解决哲学家进餐问题 | 82 |
| 2.2 进程的描述 | 39 | 3.5 进程通信 | 84 |
| 2.2.1 进程的定义与特征 | 39 | | |
| 2.2.2 进程的基本状态 | 40 | | |
| 2.2.3 进程的挂起状态 | 42 | | |

| | | | | | |
|------------------|--------------------|-----|------------------|-----------------------|-----|
| 3.5.1 | 进程通信的类型 | 84 | 4.8.1 | 死锁的检测 | 128 |
| 3.5.2 | 直接通信和间接通信方式 | 85 | 4.8.2 | 死锁的解除 | 130 |
| 3.5.3 | 消息传递系统中的几个问题 | 87 | 习题 | | 131 |
| 3.5.4 | 消息缓冲队列通信机制 | 89 | 第5章 存储器管理 | | 133 |
| 3.6 | OS/2 进程同步和进程通信 | 91 | 5.1 | 程序的装入和链接 | 133 |
| 3.6.1 | OS/2 进程同步机制 | 91 | 5.1.1 | 程序的装入 | 133 |
| 3.6.2 | OS/2 进程通信机制 | 92 | 5.1.2 | 程序的链接 | 136 |
| 习题 | | 94 | 5.2 | 连续分配存储管理方式 | 137 |
| 第4章 调度与死锁 | | 96 | 5.2.1 | 单一连续分配 | 138 |
| 4.1 | 调度的类型和模型 | 96 | 5.2.2 | 固定分区分配 | 139 |
| 4.1.1 | 调度类型 | 96 | 5.2.3 | 动态分区分配 | 140 |
| 4.1.2 | 调度队列模型 | 98 | 5.2.4 | 动态重定位分区分配 | 144 |
| 4.1.3 | 选择调度方式和算法的 若干准则 | 100 | 5.2.5 | IBM-PC 微机中的 存储管理方式 | 146 |
| 4.2 | 调度算法 | 101 | 5.3 | 对换 | 147 |
| 4.2.1 | 先来先服务调度算法 | 101 | 5.3.1 | 多道程序环境下的对换 | 147 |
| 4.2.2 | 短作业(进程)优先调度算法 | 103 | 5.3.2 | 对换空间的管理 | 148 |
| 4.2.3 | 时间片轮转调度算法 | 103 | 5.3.3 | 进程的换出与换入 | 148 |
| 4.2.4 | 优先权调度算法 | 105 | 5.4 | 分页存储管理方式 | 149 |
| 4.2.5 | 高响应比优先调度算法 | 106 | 5.4.1 | 分页存储管理的基本方法 | 150 |
| 4.2.6 | 多级队列调度 | 107 | 5.4.2 | 地址变换机构 | 151 |
| 4.2.7 | 多级反馈队列调度算法 | 107 | 5.4.3 | 两级和多级页表 | 154 |
| 4.3 | 实时系统中的调度 | 109 | 5.4.4 | 反置页表 | 156 |
| 4.3.1 | 对实时系统的要求 | 109 | 5.5 | 分段存储管理 | 157 |
| 4.3.2 | 实时调度算法 | 110 | 5.5.1 | 分段存储管理方式的引入 | 157 |
| 4.3.3 | 实时调度实例 | 110 | 5.5.2 | 分段系统的基本原理 | 158 |
| 4.4 | 多处理机调度 | 113 | 5.5.3 | 共享与保护 | 161 |
| 4.4.1 | 进程调度 | 113 | 5.5.4 | 段页式存储管理方式 | 162 |
| 4.4.2 | 自调度 | 114 | 习题 | | 164 |
| 4.4.3 | 成组调度 | 115 | 第6章 虚拟存储器 | | 165 |
| 4.4.4 | 专用处理机分配 | 116 | 6.1 | 虚拟存储器的基本概念 | 165 |
| 4.5 | OS/2 调度 | 117 | 6.1.1 | 虚拟存储器的引入 | 165 |
| 4.5.1 | 线程的优先权 | 117 | 6.1.2 | 虚拟存储器的实现方式 | 167 |
| 4.5.2 | 调度算法 | 118 | 6.1.3 | 虚拟存储器的特征 | 167 |
| 4.6 | 死锁的基本概念 | 119 | 6.2 | 请求分页存储管理方式 | 168 |
| 4.6.1 | 产生死锁的原因 | 119 | 6.2.1 | 请求分页中的硬件支持 | 168 |
| 4.6.2 | 产生死锁的必要条件 | 121 | 6.2.2 | 页面分配 | 171 |
| 4.6.3 | 处理死锁的基本方法 | 122 | 6.2.3 | 页面调入策略 | 172 |
| 4.7 | 死锁的预防和避免 | 122 | 6.3 | 页面置换算法 | 174 |
| 4.7.1 | 死锁的预防 | 123 | 6.3.1 | 最佳置换算法和先进先出算法 | 174 |
| 4.7.2 | 系统的安全状态 | 124 | 6.3.2 | 最近最久未使用 LRU 置换算法 | 176 |
| 4.7.3 | 利用银行家算法避免死锁 | 125 | 6.3.3 | Clock 置换算法 | 177 |
| 4.8 | 死锁的检测和解除 | 128 | | | |

| | | | |
|---|-----|--------------------------|-----|
| 6.3.4 其它置换算法 | 179 | 第8章 文件系统 | 226 |
| 6.4 请求分页系统的性能分析 | 180 | 8.1 文件和文件系统 | 226 |
| 6.4.1 缺页率对有效访问时间的影响 | 180 | 8.1.1 文件、记录和数据项 | 226 |
| 6.4.2 工作集 | 181 | 8.1.2 文件类型 | 228 |
| 6.4.3 抖动产生的原因和预防方法 | 182 | 8.1.3 文件系统模型 | 229 |
| 6.5 请求分段存储管理方式 | 184 | 8.1.4 文件操作 | 230 |
| 6.5.1 请求分段中的硬件支持 | 184 | 8.2 文件逻辑结构 | 232 |
| 6.5.2 分段共享与保护 | 186 | 8.2.1 文件逻辑结构的类型 | 232 |
| 6.6 OS/2 存储器管理 | 188 | 8.2.2 顺序文件 | 233 |
| 6.6.1 Intel 80386 的寻址方式和 存储器管理方式 | 189 | 8.2.3 索引文件 | 234 |
| 6.6.2 分段存储管理 | 190 | 8.2.4 索引顺序文件 | 235 |
| 6.6.3 分页存储管理 | 192 | 8.3 目录管理 | 236 |
| 习题 | 194 | 8.3.1 文件控制块和索引结点 | 237 |
| 第7章 设备管理 | 196 | 8.3.2 单级目录结构 | 239 |
| 7.1 I/O系统的组成 | 196 | 8.3.3 两级目录结构 | 240 |
| 7.1.1 I/O系统的结构 | 196 | 8.3.4 树型目录结构 | 241 |
| 7.1.2 I/O设备 | 197 | 8.3.5 目录查询技术 | 243 |
| 7.1.3 设备控制器 | 198 | 8.4 文件共享 | 244 |
| 7.1.4 I/O通道 | 200 | 8.4.1 早期实现文件共享的方法 | 244 |
| 7.2 I/O控制方式 | 202 | 8.4.2 基于索引结点的共享方式 | 245 |
| 7.2.1 程序I/O方式 | 203 | 8.4.3 利用符号链实现文件共享 | 247 |
| 7.2.2 中断驱动I/O控制方式 | 204 | 8.5 文件保护 | 247 |
| 7.2.3 直接存储器访问DMA 控制方式 | 204 | 8.5.1 保护域 | 248 |
| 7.2.4 I/O通道控制方式 | 206 | 8.5.2 访问矩阵 | 249 |
| 7.3 缓冲管理 | 207 | 8.5.3 访问矩阵的修改 | 250 |
| 7.3.1 缓冲的引入 | 207 | 8.5.4 访问矩阵的实现 | 251 |
| 7.3.2 单缓冲 | 208 | 8.5.5 分级安全管理 | 253 |
| 7.3.3 双缓冲 | 209 | 习题 | 256 |
| 7.3.4 循环缓冲 | 209 | 第9章 磁盘存储器管理 | 257 |
| 7.3.5 缓冲池 | 211 | 9.1 磁盘I/O | 257 |
| 7.4 设备分配 | 213 | 9.1.1 磁盘性能简述 | 257 |
| 7.4.1 设备分配中的数据结构 | 213 | 9.1.2 早期的磁盘调度算法 | 260 |
| 7.4.2 设备分配时应考虑的若干因素 | 215 | 9.1.3 各种扫描算法 | 261 |
| 7.4.3 设备独立性 | 216 | 9.2 外存分配方法 | 263 |
| 7.4.4 独占设备的分配程序 | 218 | 9.2.1 连续分配 | 263 |
| 7.4.5 SPOOLing 技术 | 219 | 9.2.2 链接分配 | 264 |
| 7.5 设备处理 | 221 | 9.2.3 索引分配 | 267 |
| 7.5.1 设备驱动程序的功能和特点 | 221 | 9.3 空闲存储空间的管理 | 270 |
| 7.5.2 设备驱动程序的处理过程 | 222 | 9.3.1 空闲表法 | 270 |
| 7.5.3 中断处理程序的处理过程 | 223 | 9.3.2 空闲链表法 | 270 |
| 习题 | 225 | 9.3.3 位示图法 | 271 |
| | | 9.3.4 成组链接法 | 272 |
| | | 9.4 磁盘容错技术 | 273 |

| | | | |
|----------------------------------|-----|-------------------------------------|-----|
| 9.4.1 第一级容错技术 | 274 | 11.2.3 服务存取点和连接 | 333 |
| 9.4.2 第二级容错技术 | 275 | 11.2.4 OSI 七层模型 | 334 |
| 9.4.3 廉价磁盘冗余阵列 | 276 | 11.3 客户/服务器模式和对等模式 | 335 |
| 9.4.4 后备系统 | 277 | 11.3.1 客户/服务器模式的 形成和结构 | 335 |
| 9.5 文件系统性能的改善 | 279 | 11.3.2 客户/服务器模式的类型 | 336 |
| 9.5.1 磁盘高速缓存 | 280 | 11.3.3 客户/服务器间的交互 | 336 |
| 9.5.2 优化数据的分布 | 281 | 11.3.4 客户/服务器模式的优点 | 337 |
| 9.5.3 提高磁盘 I/O 速度的其它方法 | 282 | 11.3.5 对等模式 | 337 |
| 9.6 数据一致性控制 | 283 | 11.4 网络操作系统的构成 | 338 |
| 9.6.1 事务 | 284 | 11.4.1 工作站网络软件 | 338 |
| 9.6.2 检查点 | 285 | 11.4.2 网络环境软件 | 340 |
| 9.6.3 并发控制 | 286 | 11.4.3 网络服务软件 | 341 |
| 9.6.4 若干具体的数据一致性问题 | 286 | 11.4.4 网络管理软件 | 341 |
| 习题 | 289 | 11.5 文件与打印服务 | 342 |
| 第 10 章 操作系统接口 | 291 | 11.5.1 文件服务的引入 | 342 |
| 10.1 联机命令接口 | 291 | 11.5.2 文件服务软件 | 344 |
| 10.1.1 命令格式 | 291 | 11.5.3 打印服务 | 345 |
| 10.1.2 联机命令的类型 | 293 | 11.6 电子邮件服务 | 347 |
| 10.1.3 终端处理程序 | 296 | 11.6.1 电子邮件系统 | 347 |
| 10.1.4 命令解释程序 | 298 | 11.6.2 报文处理系统的引入 | 349 |
| 10.2 程序接口 | 300 | 11.6.3 报文处理系统的组成 | 350 |
| 10.2.1 系统调用的基本概念 | 300 | 习题 | 351 |
| 10.2.2 系统调用的类型 | 302 | 第 12 章 分布式操作系统 | 352 |
| 10.2.3 系统调用的实现 | 304 | 12.1 进程迁移 | 353 |
| 10.3 图形用户接口元素 | 305 | 12.1.1 计算和数据的迁移 | 353 |
| 10.3.1 窗口 | 306 | 12.1.2 引入进程迁移的原因 | 354 |
| 10.3.2 图标 | 308 | 12.1.3 进程迁移机制 | 354 |
| 10.3.3 菜单 | 310 | 12.1.4 迁移的协商 | 356 |
| 10.3.4 对话框 | 311 | 12.2 分布式进程同步 | 357 |
| 10.4 图形用户接口元素的基本操作 | 313 | 12.2.1 事件排序 | 358 |
| 10.4.1 菜单操作 | 313 | 12.2.2 Lamport 算法 | 358 |
| 10.4.2 窗口操作 | 315 | 12.2.3 Ricart and Agrawala 算法 | 359 |
| 10.4.3 对话框操作 | 317 | 12.2.4 令牌传送法 | 360 |
| 习题 | 319 | 12.3 分布式进程通信 | 361 |
| 第 11 章 网络操作系统的构成和服务 | 321 | 12.3.1 数据传送 | 361 |
| 11.1 计算机网络概述 | 321 | 12.3.2 路由选择 | 362 |
| 11.1.1 计算机网络的组成 | 321 | 12.3.3 流量控制 | 363 |
| 11.1.2 计算机网络的拓扑结构 | 323 | 12.3.4 分布式消息传递方式 | 365 |
| 11.1.3 交换网 | 326 | 12.3.5 远程过程调用 | 366 |
| 11.1.4 局域网 | 328 | 12.4 分布式系统中的死锁 | 368 |
| 11.2 开放系统互连参考模型 | 330 | 12.4.1 死锁的类型 | 368 |
| 11.2.1 网络体系结构的基本概念 | 330 | 12.4.2 死锁检测中的问题 | 370 |
| 11.2.2 对等实体间的通信 | 331 | | |

| | | | | | |
|----------------------------|---------------------------|-----|-------------------------------|---------------------------|-----|
| 12.4.3 | 死锁的预防 | 371 | 14.1.3 | 块设备缓冲区的分配与回收 | 425 |
| 12.4.4 | 分布式死锁的检测 | 373 | 14.2 | 设备处理程序 | 428 |
| 12.5 | 任务分配 | 374 | 14.2.1 | 核心与驱动程序的接口 | 428 |
| 12.5.1 | 任务分配时应考虑的问题 | 374 | 14.2.2 | 磁盘驱动程序 | 429 |
| 12.5.2 | 最早调度算法 | 376 | 14.2.3 | 磁盘读、写程序 | 430 |
| 12.5.3 | 子集调度算法 | 377 | 14.3 | 对文件的管理 | 431 |
| 12.5.4 | 搜索树 | 378 | 14.3.1 | 文件存储空间的管理 | 432 |
| 12.6 | 网络和分布式系统的安全性 | 380 | 14.3.2 | 文件的物理结构 | 433 |
| 12.6.1 | 对网络和分布式系统 安全性的威胁 | 381 | 14.3.3 | 用户文件描述符表和 文件表的管理 | 435 |
| 12.6.2 | 数据加密的基本概念 | 382 | 14.4 | 目录管理 | 436 |
| 12.6.3 | 数据加密标准和数字签名 | 384 | 14.4.1 | 对索引结点的管理 | 437 |
| 12.6.4 | 网络加密 | 386 | 14.4.2 | 构造目录和删除目录 | 438 |
| 12.6.5 | 密钥分配 | 388 | 14.4.3 | 检索目录 | 439 |
| 习题 | | 389 | 14.5 | 文件系统的系统调用 | 440 |
| 第 13 章 UNIX 进程控制子系统 | | 390 | 14.5.1 | 系统调用 open | 440 |
| 13.1 | UNIX 系统概述 | 390 | 14.5.2 | 系统调用 creat | 442 |
| 13.1.1 | UNIX 系统的发展 | 390 | 14.5.3 | 系统调用 close | 443 |
| 13.1.2 | UNIX 系统的特性 | 391 | 14.5.4 | 系统调用 link 和 unlink | 443 |
| 13.1.3 | UNIX 系统核心的结构 | 392 | 14.5.5 | 系统调用 read 和 write | 444 |
| 13.2 | 进程的描述 | 394 | 14.5.6 | 其它系统调用 | 446 |
| 13.2.1 | 进程的数据结构 | 394 | 习题 | | 446 |
| 13.2.2 | 进程状态及其转换 | 396 | 第 15 章 Windows 操作系统的使用 | | 448 |
| 13.2.3 | 进程映像 | 397 | 15.1 | 程序管理器 | 448 |
| 13.3 | 进程的控制与调度 | 399 | 15.1.1 | 程序管理器窗口中的 应用程序组 | 449 |
| 13.3.1 | 进程控制 | 399 | 15.1.2 | 程序管理器的功能 | 450 |
| 13.3.2 | 进程调度 | 402 | 15.1.3 | 程序组和程序项的建立 | 451 |
| 13.3.3 | 进程切换 | 402 | 15.1.4 | 传送和拷贝程序项 | 453 |
| 13.4 | 进程的同步与通信 | 405 | 15.1.5 | 启动应用程序 | 453 |
| 13.4.1 | 信号 | 405 | 15.2 | 文件管理器 | 454 |
| 13.4.2 | 管道 | 407 | 15.2.1 | 目录窗口的组成 | 455 |
| 13.4.3 | 消息 | 409 | 15.2.2 | 文件管理器的功能 | 456 |
| 13.4.4 | 共享存储区 | 412 | 15.2.3 | 对文件和目录的操作 | 457 |
| 13.4.5 | 信号量 | 414 | 15.2.4 | 改变目录窗口中的显示信息 | 460 |
| 13.5 | 存储器管理 | 416 | 15.2.5 | 软盘的维护 | 461 |
| 13.5.1 | 请求调页管理的数据结构 | 416 | 15.3 | 绘画应用程序 | 462 |
| 13.5.2 | 换页进程 | 418 | 15.3.1 | 画笔窗口中的主要元素 | 463 |
| 13.5.3 | 请求调页 | 420 | 15.3.2 | 使用绘画工具作画 | 464 |
| 习题 | | 421 | 15.3.3 | 在图画中输入文本 | 466 |
| 第 14 章 UNIX 文件子系统 | | 423 | 15.3.4 | 编辑图画 | 468 |
| 14.1 | 设备缓冲管理 | 423 | 15.4 | 书写器 | 470 |
| 14.1.1 | 字符设备缓冲管理 | 423 | 15.4.1 | 编辑文档 | 470 |
| 14.1.2 | 块设备缓冲队列的结构 | 424 | | | |

| | | | | | |
|--------|--------------------|-----|--------|------------------|-----|
| 15.4.2 | 在书写器文档中嵌入图画 | 473 | 15.5.4 | 设置键盘、鼠标和端口 | 486 |
| 15.4.3 | 字符格式化 | 474 | 15.6 | 打印管理器 | 488 |
| 15.4.4 | 段落格式化 | 476 | 15.6.1 | 安装和配置打印机 | 488 |
| 15.4.5 | 整个文档格式化 | 478 | 15.6.2 | 打印管理器的功能 | 491 |
| 15.5 | 控制面板 | 481 | 15.6.3 | 应用程序中的打印功能 | 493 |
| 15.5.1 | 选配桌面颜色、图案和壁纸 | 481 | 习题 | | 495 |
| 15.5.2 | 屏幕保护 | 484 | 参考文献 | | 497 |
| 15.5.3 | 设置系统日期和时间 | 485 | | | |

第 1 章

操作系统引论

计算机由硬件和软件两部分组成，操作系统 OS(Operating System)是配置在计算机硬件上的第一层软件，是对硬件系统的第一次扩充。它在计算机系统中占据了特殊重要的地位，其它所有的软件如汇编程序、编译程序、数据库管理系统等系统软件以及大量的应用软件，都将依赖于操作系统的支持，取得它的服务。操作系统已成为从大型机直至微机都必须配置的软件。

1.1 操作系统的目标、作用和模型

在计算机系统上所配置的操作系统，其主要目标与计算机系统的规模和操作系统的类型有关，而操作系统在计算机系统所起的作用可以从不同的角度来观察。本节还将从层次结构上来介绍操作系统的模型。

1.1.1 操作系统的目标

目前存在着多种类型的 OS，不同类型的 OS 其目标各有所侧重。在计算机硬件上配置的 OS 的目标有以下几点：

1. 方便性

配置 OS 后可使计算机系统更容易使用。一个未配置 OS 的计算机系统是极难使用的，因为计算机硬件只能识别 0 和 1 这样的机器代码。因此，用户要在计算机上运行自己所编写的程序，就必须用机器语言书写程序；用户要想输入数据或打印数据，也都必须由自己用机器语言来书写相应的输入程序或打印程序。如果我们在计算机硬件上配置了 OS，用户便可通过 OS 所提供的各种命令来使用计算机系统。比如，用编译命令可方便地把用户用高级语言书写的程序翻译成机器代码，大大地方便了用户，从而使计算机变得易学易用。

2. 有效性

在未配置 OS 的计算机系统中，诸如 CPU、I/O 设备等各类资源，都会经常处于空闲状态而得不到充分利用；内存及外存中所存放的数据由于无序而浪费了存储空间。配置了 OS 后，可使 CPU 和 I/O 设备由于能保持忙碌状态而得到更为有效的利用，且由于使内存和外存中存放的数据有序而节省了存储空间。此外，OS 还可以通过合理地组织计算机的工作流

程，从而可进一步改善系统的资源利用率及提高系统的吞吐量。

方便性和有效性是操作系统最重要的两个目标。在过去很长一段时间内，由于计算机系统非常昂贵，因而使其有效性比方便性更为重要。正因如此，现在的大多数操作系统理论都着重于如何提高计算机系统资源的利用率和系统吞吐量问题。但是，近10年来在微机上所配置的操作系统，则更重视其方便性。

3. 可扩充性

随着 VLSI 技术和计算机技术的迅速发展，计算机硬件和体系结构也随之得到迅速发展，它们对 OS 提出了更高的功能和性能要求。因此，OS 必须具有很好的可扩充性方能适应发展要求。这就是说，OS 应采用模块化结构，以便于增加新的功能模块和修改老的功能模块。

4. 开放性

80 年代和 90 年代陆续出现了各种类型的计算机硬件系统。为使出自不同厂家的计算机及其设备，能通过网络加以集成化并能正确、有效地协同工作，实现应用程序的可移植性和互操作性，因而要求具有统一的开放的环境，其中首先是要求 OS 具有开放性。

1.1.2 操作系统的作用

可以从不同的观点(角度)来观察 OS 的作用。从一般用户的观点，可把 OS 看做是用户与计算机硬件系统之间的接口；从资源管理观点，则可把 OS 视为计算机系统资源的管理者。

一、OS 作为用户与计算机硬件系统之间的接口

OS 作为用户与计算机硬件系统之间接口的含义是：OS 处于用户与计算机硬件系统之间，用户通过 OS 来使用计算机系统。或者说，用户在 OS 的帮助下能够方便、快捷、安全、可靠地操纵计算机硬件和运行自己的程序。应当注意，OS 是一个系统软件，这种接口因而是软件接口，图 1-1 是 OS 作为接口的示意图。由图可以看出，用户可以通过以下两种方式来使用计算机。

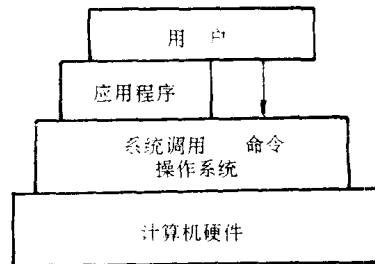


图 1-1 OS 作为接口的示意图

(1) 命令方式。这是指由 OS 提供了一组联机命令(语言)，用户可通过键盘键入有关的命令，来直接操纵计算机系统。

(2) 系统调用方式。OS 提供了一组系统调用，用户可在应用程序中通过调用相应的系统调用来操纵计算机。

二、OS 作为计算机系统资源的管理者

在一个计算机系统中，通常都包含了各种各样的硬件和软件资源。归纳起来可将资源分为四类：处理器、存储器、I/O 设备以及信息(数据和程序)。相应地，OS 的主要功能也正是针对这四类资源进行有效的管理，即：

- (1) 处理机管理。用于分配和控制处理机；
- (2) 存储器管理。主要负责内存的分配与回收；
- (3) I/O 设备管理。负责 I/O 设备的分配与操纵；
- (4) 文件管理。负责文件的存取、共享和保护。

可见，OS 确是计算机系统的资源管理者。事实上，当今世界上广为流行的一个关于 OS 作用的观点，正是把 OS 作为计算机系统的资源管理者。

三、OS 用作扩充机器

对于一台完全无软件的计算机系统(裸机)，即使其功能再强，也必定是难于使用的。如果我们在裸机上覆盖上一层 I/O 设备管理软件，用户便可利用它所提供的 I/O 命令，来进行数据输入和打印输出。此时用户所看到的机器，将是一台比裸机功能更强、使用更方便的机器。通常把覆盖了软件的机器称为扩充机器或虚机器。如果我们又在第一层软件上再覆盖上一层文件管理软件，则用户可利用该软件提供的文件存取命令，来进行文件的存取。此时，用户所看到的是一台功能更强的虚机器。如果我们又在文件管理软件上再覆盖上一层面向用户的窗口软件，则用户便可在窗口环境下方便地使用计算机，形成一台功能极强的虚机器。

由此可知，每当人们在计算机系统上覆盖上一层软件后，系统功能便增强一级。由于 OS 自身包含了若干层软件，因此当在裸机上覆盖上 OS 后，便可获得一台功能显著增强，使用极为方便的多层扩充机器或多层虚机器。

1.1.3 操作系统的层次模型

近年来，大型软件都是采用层次式结构，亦即将一个软件分为若干个层次。我们也可以利用一个层次式的 OS 模型来描述 OS。该模型分为三个层次，如图 1-2 所示。其最底层是 OS 对象，中间层是对对象进行操纵和管理的软件集合，其最高层是 OS 提供给用户使用的用户接口。

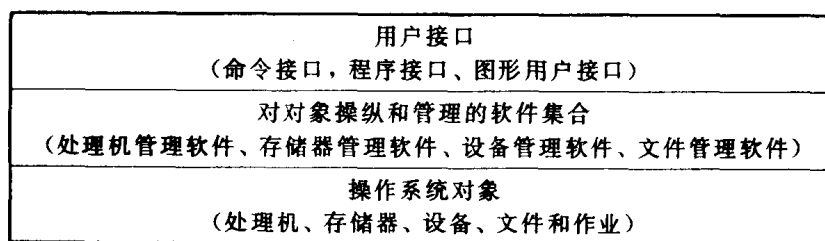


图 1-2 OS 层次模型

1. OS 对象

这就是 OS 操纵和管理的对象。由于 OS 主要用于管理计算机系统中的各种硬件和软件资源，故 OS 的对象有以下几类：一个或多个处理器，存储器系统，各类 I/O 设备以及文件和作业。

2. 对对象操纵和管理的软件集合

通常该软件集合是操作系统的核心部分，它们集中了操作系统五大功能中的四大功能，分别用于对上述四类对象进行操纵和管理。此外，还包括了一部分对作业进行管理的

功能。

3. 用户接口

为方便用户使用 OS, 利用 OS 所提供的各种功能和服务, OS 通常向用户提供以下三种类型的接口:

(1) 命令接口。这是用户和 OS 的接口。用户可以直接从键盘终端键入某种命令来取得 OS 的服务;

(2) 程序接口。这是应用程序与 OS 的接口。用户通过在程序中安排系统调用, 来取得 OS 的服务;

(3) 图形用户接口。在 90 年代推出的新型 OS, 普遍配置了更为直观的图形用户接口, 并将系统的各项功能及各种应用程序, 都以各种形式的图标逼真的表示出来, 再利用鼠标进行操作, 这样可使计算机的操作更为简单方便、生动有趣。

1.2 操作系统的发展过程

OS 的形成迄今已有约 40 年的时间。在 50 年代中期出现了第一个简单的批处理操作系统。到 60 年代中期产生了多道程序批处理系统, 不久又出现了基于多道程序的分时系统。80 年代是微机和计算机局域网大发展的年代, 同时也是微机 OS 和 LAN OS 的形成和大发展的年代。

1.2.1 推动操作系统发展的主要动力

在短短的 40 年中, 操作系统取得了如此重大的进展, 其主要动力可归结为以下四个方面。

1. 不断提高计算机资源利用率的需要

在计算机发展的初期, 计算机系统特别昂贵, 人们必须千方百计地提高计算机系统中各种资源的利用率, 这就成为推动 OS 发展的动力。由此形成了能自动地对一批作业进行处理的批处理系统。

2. 方便用户

当资源利用率不高的问题得到基本解决后, 用户(主要是程序员)在上机、调试程序时的不方便性便成为主要矛盾。于是人们又想方设法改善用户上机、调试程序时的条件, 这又成为继续推动 OS 发展的主要因素(动力)。随之便形成了允许进行人机交互的分时系统, 或称为多用户系统。

3. 器件的不断更新换代

计算机器件在不断地更新, 由第一代的电子管发展到第二代的晶体管、第三代的集成电路、第四代的大规模集成电路, 使得计算机的性能不断提高, 其规模也在急剧扩大, 从而推动着 OS 的功能和性能也迅速提高。例如, 当微机由 8 位发展到 16 位, 进而发展到 32 位时, 相应的微机 OS 也就由 8 位微机 OS 发展到 16 位, 进而发展到 32 位微机 OS。与此相应, OS 的功能和性能都有显著的提高。

4. 计算机体系结构的不断发展

计算机体系结构的发展也不断地推动着 OS 的发展并产生新的操作系统类型。例如，当计算机由单处理机系统发展为多处理机系统时，相应地，操作系统也就由单处理机 OS 发展为多处理机 OS。又如，当计算机继续发展而出现了计算机网络后，相应地，也就有了网络操作系统。

1.2.2 无操作系统时的计算机系统

一、人工操作方式

从第一代计算机诞生(1945 年)到 50 年代中期的计算机尚属第一代，还未出现 OS。这时的计算机操作是由用户(即程序员)采用人工操作方式直接使用计算机硬件系统，即由程序员将事先已穿孔(对应于程序和数据)的纸带(或卡片)装入纸带输入机(或卡片输入机)，再启动它们将程序和数据输入计算机，然后启动计算机运行。当程序运行完毕并取走计算结果后，才让下一个用户上机。这种人工操作方式有以下两个缺点。

(1) 用户独占全机。一台计算机的全部资源只能由一个用户独占。

(2) CPU 等待人工操作。当用户进行装带(卡)、卸带(卡)等人工操作时，CPU 是空闲的。

可见，人工操作方式严重降低了计算机资源的利用率，此即所谓的人工操作方式与机器利用率的矛盾，简称人机矛盾。随着 CPU 速度的提高、系统规模的扩大，人机矛盾也就变得日趋严重。此外，随着 CPU 速度的迅速提高而 I/O 设备的速度却提高缓慢，又使 CPU 与 I/O 设备之间速度不匹配的矛盾更加突出。为了缓和此矛盾，先后出现了通道技术、缓冲技术，但却未能很好地解决上述矛盾，而后来引入的脱机输入输出方式才获得了较为令人满意的结果。

二、脱机输入输出方式(Off-Line I/O)

为了解决人机矛盾及 CPU 和 I/O 设备之间速度不匹配的矛盾，50 年代末出现了脱机输入输出技术。该技术是指事先将装有用户程序和数据的纸带(或卡片)装入纸带(或卡片)输入机，在一台外围机的控制下把纸带(卡片)上的数据(程序)输入到磁带上。当 CPU 需要这些程序和数据时再从磁带上高速地调入内存。

类似地，当 CPU 需要输出时可由 CPU 直接高速地把数据从内存送到磁带上，然后再在另一台外围机的控制下，将磁带上的结果通过相应的输出设备输出。图 1-3 示出了脱机输入输出过程。由于程序和数据的输入和输出都是在外围机的控制下完成的，或者说它们是在脱离主机的情况下进行的，故称为脱机输入输出。

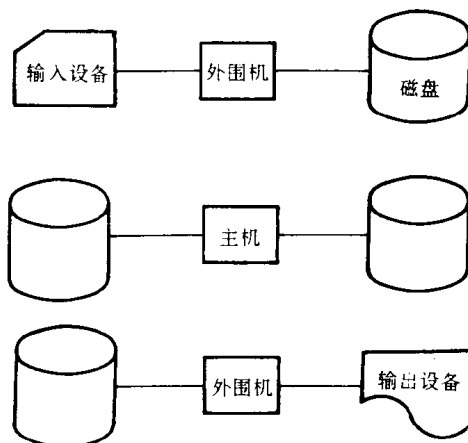


图 1-3 脱机 I/O 示意图

出方式；反之，在主机的直接控制下进行输入输出的方式称为联机输入输出 (On-Line I/O) 方式。这种脱机 I/O 方式的主要优点如下：

(1) 减少了 CPU 的空闲时间。装带(卡)、卸带(卡)、以及将数据从低速 I/O 设备送到高速的磁带(或盘)上，都是在脱机情况下进行的，它们不占用主机时间，从而有效地减少了 CPU 的空闲时间，缓和了人机矛盾。

(2) 提高 I/O 速度。当 CPU 在运行中需要数据时，是直接从高速的磁带或磁盘上将数据调入内存的，不再是从低速 I/O 设备上输入，从而大大缓和了 CPU 和 I/O 设备速度不匹配的矛盾，进一步减少了 CPU 的空闲时间。

1.2.3 单道批处理系统

一、批处理系统的处理过程

早期的计算机系统非常昂贵，为了能充分地利用它，应尽量让该系统连续地运行，以减少空闲时间。为此，通常是把一批作业以脱机输入方式输入到磁带上，并在系统中配上监督程序 (Monitor)，在它的控制下使这批作业能一个接一个地连续处理。其自动处理过程是：首先，由监督程序将磁带上的第一个作业装入内存，并把运行控制权交给该作业。当该作业的处理完成时又把控制权交还给监督程序，再由监督程序把磁带(盘)上的第二个作业调入内存。计算机系统就这样自动地一个作业一个作业地进行处理，直至磁带(盘)上的所有作业全部完成，这样便形成了早期的批处理系统。由于系统对作业的处理都是成批地进行的、且在内存中始终只保持一道作业，故称为单道批处理系统 (Simple Batch System)。图 1-4 示出了单道批处理系统的处理流程。

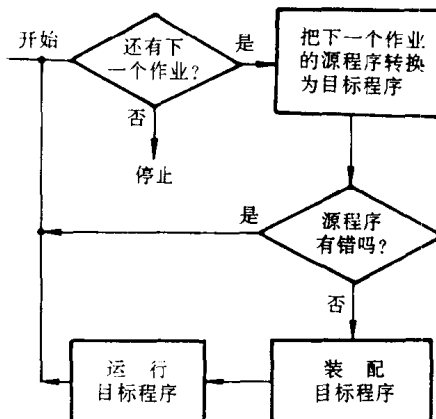


图 1-4 单道批处理系统的处理流程

由上所述不难看出，单道批处理系统是在解决人机矛盾和 CPU 与 I/O 设备速率不匹配的矛盾的过程中形成的。换言之，批处理系统旨在提高系统资源的利用率和系统吞吐量。但这种单道批处理系统仍然不能很好地利用系统资源，故现在已很少使用。

二、单道批处理系统的特征

单道批处理系统是最早出现的一种 OS，严格地说，它只能算作是 OS 的前身而并非是目前人们所理解的 OS。尽管如此，该系统比起人工操作方式已有很大的进步。该系统的主要特征如下：

- (1) 自动性。在顺利的情况下，在磁带上的一批作业能自动地逐个作业依次运行，而无须人工干预。
- (2) 顺序性。磁带上的各道作业是顺序地进入内存，各道作业完成的顺序与它们进入