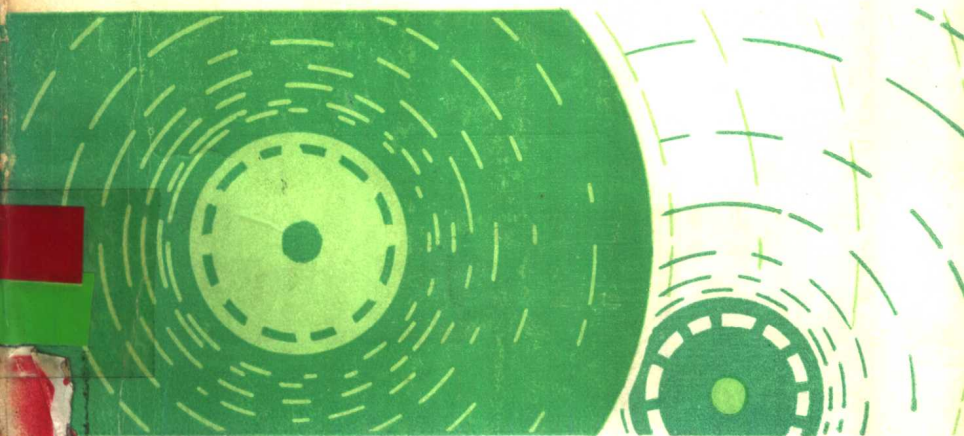


计算机网络 操作系统

王敬觉 编著



湖南大学出版社

计算机网络操作系统

王敬觉 编著

湖南大学出版社

内 容 简 介

本书专门论述以计算机网络为基础的分布式操作系统的结构和实现技术。全书共分10章。其中,前5章论述了系统的结构模型,命名,并行控制,分布事务,以及可靠性问题等基本理论和基本技术;第七章论述了分布式程序设计语言的结构和实现;其余各章分别论述了名字服务器、文件服务器的设计,以及两个分布式操作系统样机的结构。

本书可以作为计算机高年级本科生和研究生的教科书,也可以作为从事计算机工作的技术人员的参考书。

计算机网络操作系统

王敬觉 编著

责任编辑 朱 华



湖南大学出版社出版发行
(长沙市岳麓山)

湖南省新华书店经销

湘潭大学印刷厂印刷



787×1092 32开 10.375印张 233千字

1989年3月第1版 1989年3月第1次印刷

印数: 1—2500册

ISBN 7-314-00299-1/TP·11

定价: 2.05元

序 言

本书将论述以计算机网络为基础的分布式操作系统结构和实现。它是作者在给计算机应用专业编写分布式操作系统讲义和积多次讲授该课经验的基础上编著而成的。

如果在计算机网络上建成了一个分布式操作系统，那么虽然这个操作系统运行在网络的多个主机上，但用户所看到的却是一个运行在“虚拟单机”上的集中式操作系统，并且它具有很好的并行性、可扩充性和可用性。在此基础上，还可以建成多种分布式程序设计语言和调试工具，不但为计算机网络软件的开发提供极大的方便，而且为用户的使用也提供很友好的界面。可是，现在市场上所销售的各种类型的计算机网络，几乎都是在OSI七层协议的基础上提供一些功能性协议或服务（例如，文件传输、电子邮件、虚拟终端等），其透明性、并行性、可扩充性以及可用性均很差；根本谈不上具有分布式操作系统环境；使网络软件的开发和调试非常困难，使应用范围也受到很大限制。

因此，从70年代起世界上有不少大学和研究机构开始在计算机网络上建成分布式操作系统的理论、结构和实现技术的研究工作。本书是在收集有关这方面资料的基础上，对此进行系统的论述。

全书共分十章。

第一章，网络操作系统（NOS）的结构模型，将论述分布式操作系统的结构，它是具有指导意义的一章。

第二章，名字与命名机构，将论述分布式操作系统中对象的命名规则，和命名机构的设计。这是系统中一切操作的

QJS 103/11

基础，因为只有通过对象的名字才能对对象实施操作。

第三章，网络中的并行控制，将论述以加锁为基础的并行控制机构和以时间标记为基础的并行控制机构。

第四章，分布事务管理，将论述事务原子性和实现技术。

第五章，可靠性问题，将论述系统故障检查、容错、恢复理论和技术。

以上 5 章可以算是分布式操作系统的理论和技术基础。

为了使读者能够对分布式操作系统结构和实现技术有较为具体的了解，本书还在第六章论述了名字服务器的设计，第八章论述了文件服务器的设计，第九章论述了 NSW 分布式操作系统的结构，第十章论述了 Cronus 分布式操作系统的结构。

本书在第七章还专门论述了分布式程序设计语言的结构和实现。

由于分布式操作系统至今仍处于不断发展和完善阶段，由于作者水平有限，书中不当之处在所难免。欢迎读者予以指正。

作者在编写此书过程中曾得到国防科技大学陈火旺教授的帮助，在此表示非常的感谢。

王敬觉

1988年 6 月

目 录

第一章 网络操作系统 (NOS) 的结构模型	(1)
§1—1 引论.....	(1)
§1—2 NOS的结构模型.....	(4)
§1—3 IPC 层的结构模型.....	(7)
§1—4 服务支持层的结构模型.....	(13)
§1—5 服务器的结构模型.....	(22)
§1—6 对象的共享.....	(32)
§1—7 分布文件系统的 HPC 模型.....	(33)
§1—8 关于应用层.....	(37)
第二章 名字与命名机构	(38)
§2—1 引论.....	(38)
§2—2 命名系统的目标.....	(40)
§2—3 面向使用人员的名字.....	(43)
§2—4 唯一的面向机器的名字.....	(46)
§2—5 关于寻址和寻路由的问题.....	(52)
第三章 网络中的并行控制	(56)
§3—1 引论.....	(56)
§3—2 网络中事件的排序.....	(58)
§3—3 资源共享的同步机构.....	(61)

§3—4 时钟同步机构.....	(63)
§3—5 数据库多拷贝同步更新.....	(65)
第四章 分布事务管理.....	(84)
§4—1 引论.....	(84)
§4—2 分布事务的原子性和坚持性.....	(86)
§4—3 分布事务的并行控制.....	(102)
§4—4 分布事务的结构.....	(106)
第五章 可靠性问题	(111)
§5—1 基本概念.....	(111)
§5—2 提高可靠性的非锁提交协议.....	(114)
§5—3 容错与增加事务处理能力.....	(124)
§5—4 网络状态的取得.....	(131)
§5—5 数据不一致性的检测 and 解决.....	(134)
§5—6 检查点和冷启动问题.....	(137)
第六章 名字服务器的设计.....	(141)
§6—1 引论.....	(141)
§6—2 V—系统的名字服务器设计.....	(144)
第七章 分布式程序设计语言.....	(157)
§7—1 引论.....	(157)
§7—2 Ada 语言概述.....	(159)
§7—3 子程序、程序包和任务.....	(168)
§7—4 一个简化的分布式程序设计语言CMAY.....	(186)
§7—5 分布程序设计中的原语.....	(190)

§7—6 一个CMAY 分布程序实例.....	(198)
第八章 文件服务器设计.....	(200)
§8—1 引论.....	(200)
§8—2 文件的使用和管理.....	(207)
§8—3 支持原子事务的文件服务器内部设计.....	(213)
§8—4 文件服务器对通讯的要求.....	(223)
§8—5 文件服务器实例简介.....	(228)
第九章 第二代以网络为基础的 分布式操作系统——NSW	(231)
§9—1 引论.....	(231)
§9—2 MSG 的设计	(235)
§9—3 网络与局部的接口进程 Foreman.....	(253)
§9—4 FE 和 WM 的结构.....	(273)
§9—5 NSW文件操作进程 File package(FP).....	(287)
第十章 第三代以网络为基础的 分布式操作系统——Cronus.....	(299)
§10—1 系统的拓朴结构.....	(299)
§10—2 Cronus 的对象模型	(301)
§10—3 Cronus 内核的结构	(310)
§10—4 分布式应用软件开发.....	(313)
参考文献与书目.....	(322)

第一章 网络操作系统 (NOS)的结构模型

§ 1—1 引 论

一、NOS问题的提出

ISO 的OSI七层协议并不能满足资源共享和分布计算的要求。这是因为OSI的七层协议只是些功能协议。也就是说只是实现某几种功能。例如，文件传送协议 (FTP) 只是实现文件的传送。又如电子邮件协议也只是实现邮件的传送等等。但是，它们都不能够提供一个操作系统的环境。其原因是：

1. 在现有的资源和服务的基础上，要建立新资源新服务相当困难，因为七层协议并未提供什么好的基础。如果一个程序员想提供和使用一个新的网络上能共享的资源，那么他就必须解决以下一些问题：

- 1) 数据类型的翻译问题
- 2) 命令和回答的格式和语法分析问题
- 3) 命名问题
- 4) 资源的保护问题
- 5) 出错控制问题
- 6) 资源管理问题

7) 同步机构问题

8) 和进程间通讯层 (IPC) 的接口问题

这就是说, 每当要建立一个新资源和相应的新服务时, 本来有不少公共的事务要做, 如果说这些公共事务事先做好, 那么, 建立一个新资源和新服务也就容易多了。但是 OSI 的七层协议并未提供这样的基础。

2. 由于计算机网络上的机器往往是不相同的 (异构网络), 因此用户必须知道不同的操作命令, 不同的命名规则, 以及不同的访问方式, 而这样对于用户来讲是很不方便的。对于用户来讲, 我们希望向他们提供一个虚拟单机 (虽然在物理上它实际是一个计算机网络), 但 OSI 却不能做到这一点。例如, 用户只在本地主机上, 并不能知道全网络上资源的分布。

3. OSI 的七层协议不能实现任务的分解, 负载的平衡, 动态的调度, 也不易实现在全网范围内对用户使用资源进行记帐管理。

正是由于以上原因, 70年代初, 人们就开始着手考虑是否能够在网络基础上建立一个计算机网络操作系统的问题, 并且随着网络技术和网络应用的普及, NOS 的迫切性更增加了。现在已经研制出了多个 NOS 的样板系统, 例如:

1) RSEXEC 系统: 这是一个资源共享执行系统 (Resource Sharing Execution), 它由美国国防部高级研究工程机构 (DARPA) 所支持, 并在 70 年代早期被研制成功。

2) NSW 系统: 这是有名的国家软件公司 (National software works) 工程, 它由 DARPA 和美国空军开发中心

所支持, 74年完成方案论证, 78年完成第一期工程, 以后它还在被发展。

3) MIKE系统: 这是一个在双环计算机网络(DDLCN)上建立的一个NOS。它由美国俄亥俄大学计算机和信息科学系的Ming T. Liu教授等人研制成功(1981年)。

4) TABS: 这是一个面向事务的分布系统(Transaction based systems)。它由IBM公司给予支持, 并由卡里基——梅伦大学计算机科学系ALFRED Z. SPECTOR等人所研制。

5) Eden系统: 这是一个由美国自然科学基金资助的并由华盛顿大学所研制成功的一个系统(1981—1985年)。

6) Cronus系统: 这也是一个由美国自然科学基金资助, 并由麻省BBN实验室研制成功(81—85)。

由此看来, NOS是一个很活跃的领域, 我国现在正在推广计算机网络的应用, 已经, 并在以后必将碰到大量的NOS问题, 因此, 加强对NOS的研究很有必要。

二、NOS设计的目标

1. 一个终端用户, 一个进程或一个程序员对于网络上的资源应当有一个统一的观点。这就是说, 不管什么资源, 不管它实际上被分布在什么地方, 用户都好象感到它就在自己的主机上一样: 一样的操作命令, 一样的访问规则。

2. NOS应当具有良好的可扩充性。

1) 在现成服务基础上, 用户很容易加上新服务。例如, 从盘块的读/写服务扩充到文件的读/写服务; 再被扩充到数据库的查询/更新服务。

2) 基本的NOS结构不要建立起所有的服务, 只要把开

始时所要的服务建立，然后若再需要新服务时，再增加之。

3) 在 NOS 中，增加一个或删除一个主机系统的开销很小。

3. NOS 能把一组硬件/软件资源转换为一个抽象的对象 (object) 的集合 (进程、文件、字典、时钟、帐号等)，这说明：它支持对象的命名、访问、共享、保护、同步、出错控制，以及对象之间相互通讯。

4. NOS 能在多个进程之间对资源进行多路转换和分配。

以上便是 NOS 所要达到的目标，显然 OSI 的协议是不可能达到以上目的的。特别是只解决了网络的低层软件 (端——端协议以下) 的问题，还很难在这样的环境下开展应用。虽然这部分软件已在过去 10 多年中解决得比较好，并已做成产品，但是，端——端协议以上的高层软件，即应用的环境并没有建设得很好。而 NOS 确实是一个很好的环境，因为有了这样的环境，分布数据库系统和分布计算才能实现。

§ 1—2 NOS 的结构模型

要设计出满足以上要求的 NOS，就必然碰到比单机操作系统复杂得多的结构问题，而且有些在单机上被使用的技术现在几乎完全不能被使用了。NOS 所面临的问题如下：

1) 翻译问题：这就是网络上不同主机的不同编码和数据表示引起的翻译问题。

2) 怎样实现分布资源和分布服务问题。

3) 怎样解决复杂的故障检测，故障定位，和故障恢复的问题。

4) 怎样解决多拷贝文件或多拷贝数据库的一致性问题。

5) 怎样实现真正的分布控制问题。

6) 怎样提高系统效率问题。

面对如此复杂的问题,如果不首先根据设计目标研究出一个满足以上目标的结构模型,那是不可能进行实用系统的设计的,因此NOS的结构模型研究就成为研究NOS中很重要的课题,现在,这方面已经取得了相当重要的成果。

一、基本概念

1. 对象(Object)

1) 实对象:这是指硬件实体,例如,处理器、辅助存储器、I/O设备等。

2) 抽象对象:这是一些软实体,例如,进程、文件、目录、虚拟I/O设备、数据库、帐目、时钟等。

3) 主动对象:它可以改变自己和其它对象的表示,例如,进程。

4) 被动对象:它是其表示只能由主动对象来改变的对象,例如,虚拟I/O设备、内存、文件、数据库等等。

在NOS中定义某一对象必须做以下两件事。

1) 确定表示某一对象状态的数据结构(如文件的目录),确定表示对象体(如文件体)本身的数据结构(如记录数组)。

2) 确定在此对象上的一组操作(如读、写)。

2. 服务器(Server)

它实现对某一对象的抽象表示,其次,它还可以根据外来的请求,在此对象上实行被请求的操作,同一类型对象的不

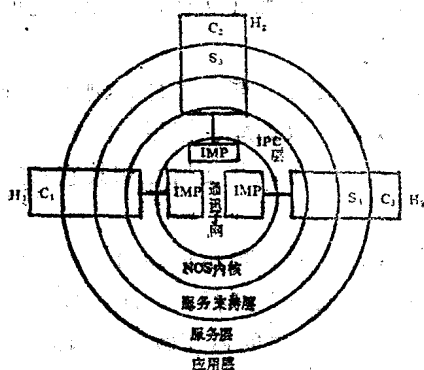
同服务器（例如，在不同主机上的文件服务器）由于在不同主机上，因此它们的内部结构细节可能相差较大，但是，从外部接口上看，它们所表示对象的状态数据结构和对象体的逻辑结构以及操作仍然是相同的。

3. 消费者（customer）：它是向实施某种服务的服务器发送请求的请求者。

4. 事务处理：一个消费者向服务器发送一个请求然后服务器又给予回答。这个处理过程便被叫做一个事务处理。

二、NOS的结构模型

NOS的层次结构如图1—1和图1—2所示。



C: 顾客 S: 服务器 H: 主机 IMP: 接口信息处理机

图 1-1 NOS的结构

1. IPC层将支持把任意长度的未被解释的消息进行传送。

2. 服务支持层：它把命名、出错控制、保护、资源管理、数据和控制信息的表示和翻译、以及同步等大多数服务

所需要的公共机构抽出来，并组成一个支持层。此外，它还定义服务操作和参数。

3. 服务层：它由一些具体的服务所组成，其中基本服务是：

文件服务、目录服务、进程服务、时钟服务等。高级服务包括数据库服务等。

4. 应用层：这是由一些应用程序所组成。

以上便是整个NOS的层次结构模型，如果要具体实现这个系统，还必须对每一层的模型进行研究，所以下面将分别论述各层的结构模型。

§ 1—3 IPC层的结构模型

一、IPC 层的接口

1. 接口上用户所能看到的抽象对象是：

1) 通讯通道(连接)：一个(源地址、目的地址)对被叫做通讯通道，或被叫做一个连接。其中源地址是源进程地址，目的地址是目的进程地址，这里的地址皆为全网标识符空间中的地址，每一个连接都有一个状态记录。这个记录包含以下几个域：

① 指向被发送信息的指针，和指向将要接收信息的缓冲区的指针。

② 指向已被成功发送和接收信息的指针。

③ 指向被怀疑出现麻烦信息的指针，并指出出现麻烦的原因。

④ 说明此连接封锁或唤醒的状态。

⑤ 其它。

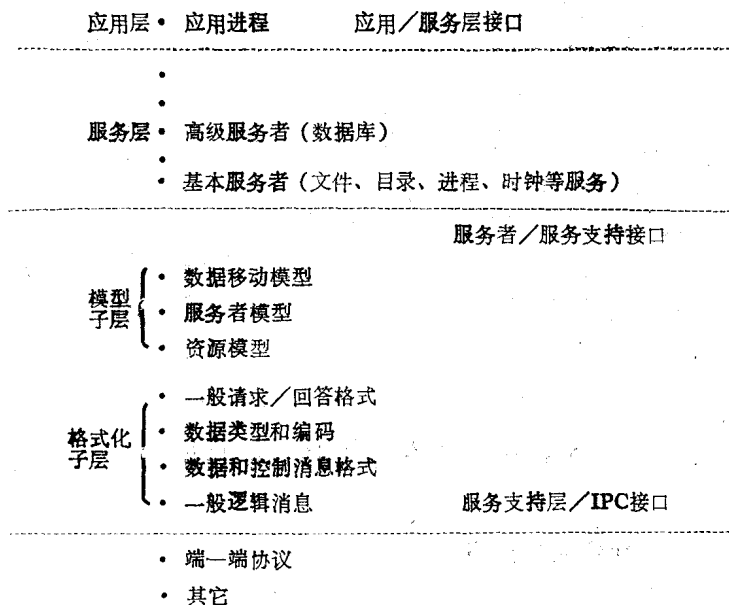


图 1-2 NOS分层结构模型

2) 数据对象:

消息: 这是以字节为单位来组成的, 其长度为任意的, 并以BOM (消息开始), 和 EOM (消息结束) 为界标的信息。一般BOM 被放在消息的第一个报文的表头中, EOM被放在最后一个报文的表头中。

2. 在接口上能够使用的操作:

1) Send: 在指定的一个连接上, 把消息从源进程发送给目的进程。

2) Receive: 表明接收者欢迎接收来自任一进程的消息。

3) Status: 它将收集有关某一连接上的状态信息。

① 什么数据已被成功地发送。

② 什么数据已被正确地接收。

③ 什么时候可以发送数据。

④ Send 和 Receive 是否遇到问题, 以及遇到什么问题。

4) Abort: 允许把一个被挂起的发送或接收删除掉, 允许停止一个正在运行的发送或接收。

5) Wait: 等待一个被挂起的发送或接收的完成, 这

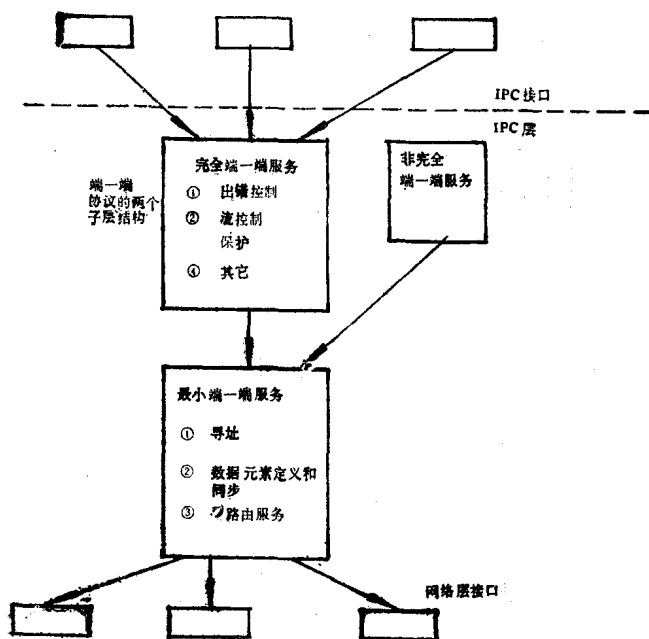


图 1-3 支持IPC服务的层次结构