第一图书网, tushu007.com

<<计算机系统设计原理>>

图书基本信息

书名：<<计算机系统设计原理>>

13位ISBN编号：9787302294597

10位ISBN编号：7302294593

出版时间：2012-12

出版时间：清华大学出版社

作者：Jerome H. Saltzer,M. Frans Kaashoek

页数：375

字数：617000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

内容概要

据我们所知，《世界著名计算机教材精选：计算机系统设计原理》在内容和方法上是非常独特的。
《世界著名计算机教材精选：计算机系统设计原理》深入而且广泛地介绍了构造计算机系统的主要原理和概念，这里我们所讲的计算机系统，是指广义的计算机系统，包括操作系统、客户端.服务器应用程序、数据库系统、安全的Web网站以及容错的磁盘集群等。
这些原则和抽象是不受时间影响的，不管读者是否是计算机系统专业的学生或专业人士，这些原则都具有重要价值。
这些原则和抽象经过了多代计算机系统的证明，《世界著名计算机教材精选：计算机系统设计原理》作者具有构建计算机系统的经验，并讲授了该课程数十年。
本书介绍了计算机系统中相当广泛的原理和抽象，并深入介绍了它们。
本书使用伪代码介绍核心概念，使得读者可以测试他们对于该概念的具体实例的理解。
通过使用伪代码，本书将客户端.服务器计算、远程过程调用、文件、线程、地址空间、尽量送达网络、原子性、验证的消息等概念的精华展现给读者。
在本书的习题中，我们延续了这一方法，使得读者可以通过研究伪代码来探讨。
本书分成两部分，第一部分是读者拿到的书籍，仅包括前6章，第二部分包括第7～11章和一些辅助材料，已发布在网络上作为开放的教育资源。
请参见后面的“到哪里找到第二部分和其他在线材料”。

作者简介

作者：（美国）索尔特（Jerome H. Saltzer） （美国）卡休克（M. Frans Kaashoek） 译者：陈文光 张广艳 陈康 薛瑞尼

书籍目录

第1章 系统
1.0 概述
1.1 系统和复杂性
1.1.1 不同领域中系统的共同问题
1.1.2 系统、组件、接口和环境
1.1.3 复杂性
1.2 复杂性的来源
1.2.1 相互影响的需求
1.2.2 保证高利用率
1.3 处理复杂性I
1.3.1 模块化
1.3.2 抽象化
1.3.3 层次化
1.3.4 分级化
1.3.5 组合起来：用名字关联
1.4 求同存异：计算机系统与其他系统的比较
1.4.1 计算机系统组合没有限制
1.4.2 d（技术）/dt是前所未有的
1.5 处理复杂性Ⅱ
1.5.1 为什么模块化、抽象化、层次化和分级化还不够
1.5.2 迭代法
1.5.3 保持简单
1.6 本书其他内容
习题
第2章 计算机系统的组成部分
2.0 概述
2.1 3种基本抽象
2.1.1 存储器
2.1.2 解释器
2.1.3 通信链路
2.2 计算机系统中的命名
2.2.1 命名模型
2.2.2 默认上下文引用和显式上下文引用
2.2.3 路径名、命名网络和递归名字解析
2.2.4 多重查找：在分层的上下文中搜索
2.2.5 名字比较
2.2.6 名字发现
2.3 用名字和层次结构组织计算机系统
2.3.1 硬件层：总线
2.3.2 软件层：文件抽象
2.4 总结经验，面向未来
2.5 案例分析：UNIX文件系统中的层次和命名
2.5.1 UNIX文件系统应用程序编程接口
2.5.2 块层
2.5.3 文件层

2.5.4 inode编号层
2.5.5 文件名层
2.5.6 路径名层
2.5.7 连接
2.5.8 重命名
2.5.9 绝对路径层
2.5.10 符号连接层
2.5.11 实现文件系统API
2.5.12 Shell，隐式上下文，搜索路径，名字发现
2.5.13 进一步阅读推荐
习题
第3章 命名方案的设计
3.0 概述
3.1 命名方案设计中的考虑因素
3.1.1 模块化共享
3.1.2 元数据与名字重载
3.1.3 地址：定位对象的名字
3.1.4 生成唯一的名字
3.1.5 预期用户与用户友好的名字
3.1.6 名字、值和绑定的相对寿命
3.1.7 回顾和展望：名字是基本的系统组件
3.2 案例研究：统一资源定位器（URL）
3.2.1 网页浏览作为参考经历；名字发现
3.2.2 URL的解释
3.2.3 URL大小写敏感性
3.2.4 部分URL的错误上下文引用
3.2.5 URL中的名字重载
3.3 战争故事：名字使用中的病症
3.3.1 名字冲突赶走了笑容
3.3.2 来自重载的脆弱名字，以及市场对策
3.3.3 来自重载的更加脆弱的名字，伴随市场崩溃
3.3.4 用户友好的名字中的大小写敏感性
3.3.5 电话号码的用尽
习题
第4章 使用客户及服务增强模块化
4.0 概述
4.1 客户/服务组织方式
4.1.1 从软模块化到强制模块化
4.1.2 客户/服务的组织方式
4.1.3 多客户端和多服务器
4.1.4 可信中间方
4.1.5 一个简单的例子服务
4.2 客户端和服务端之间的通信
4.2.1 远程过程调用（RPC）
4.2.2 RPC不等于过程调用
4.2.3 通过中间方的通信
4.3 总结及前景

4.4 案例研究：因特网域名系统（DNS）
4.4.1 DNS中的名字解析
4.4.2 层次化的名字管理
4.4.3 DNS的其他特点
4.4.4 DNS中的名字发现
4.4.5 DNS响应的可信性
4.5 案例研究：网络文件系统（NFS）
4.5.1 命名远程的文件和目录
4.5.2 NFS的远程过程调用
4.5.3 扩展UNIX文件系统来支持NFS
4.5.4 一致性
4.5.5 NFS版本3及后续版本
习题
第5章 使用虚拟化技术强制模块化
5.0 概述
5.1 在一个计算机内部使用虚拟化技术进行客户端服务器组织
5.1.1 虚拟化计算机的抽象概念
5.1.2 仿真与虚拟机
5.1.3 路线图：逐步虚拟化
5.2 使用SEND、RECEIVE以及有界缓存区的虚拟连接
5.2.1 有界限缓存区的SEND与RECEIVE的接口
5.2.2 使用有界缓存区进行顺序合作
5.2.3 竞争状态
5.2.4 锁与前后原子性
5.2.5 死锁
5.2.6 实现ACQUIRE以及RELEASE
5.2.7 使用单一写原理实现前后原子性动作
5.2.8 使用异步连接在同步岛之间进行合作
5.3 在内存上强制模块化
5.3.1 使用域强制模块化
5.3.2 使用多个域控制共享
5.3.3 使用内核态与用户态更多强制模块化
5.3.4 门与模式转换
5.3.5 为有界缓存区强制模块化
5.3.6 内核
5.4 虚拟化内存
5.4.1 虚拟化地址
5.4.2 使用页映射翻译地址
5.4.3 虚拟地址空间
5.4.4 硬件与软件对比以及旁路转换缓存
5.4.5 段（高级主题）
5.5 使用线程虚拟化处理器
5.5.1 多个线程之间共享一个处理器
5.5.2 实现YIELD
5.5.3 建立和终结线程
5.5.4 使用线程强制模块化：抢先式调度
5.5.5 使用线程和地址空间强制模块化

5.5.6 线程分层
5.6 顺序合作的线程原语
5.6.1 通知丢失问题
5.6.2 使用事件计数器以及顺序器避免通知丢失问题
5.6.3 实现AWAIT、ADVANCE、TICKET、READ（高级主题）
5.6.4 轮询、中断与顺序合作
5.7 案例分析：在Intel x86上强制模块化的演进
5.7.1 早期设计：没有对强制模块化的支持
5.7.2 使用段强制模块化
5.7.3 基于页的虚拟地址空间
5.7.4 概述：进一步的演进
5.8 应用：使用虚拟机强制模块化
5.8.1 虚拟机的使用
5.8.2 实现虚拟机
5.8.3 虚拟化的例子
习题
第6章 性能
6.0 概述
6.1 面向性能的设计
6.1.1 性能量度
6.1.2 一种系统化的面向性能的设计方法
6.1.3 利用工作负载的特性减少延迟
6.1.4 利用并发性减少延迟
6.1.5 提高吞吐率：并发性
6.1.6 排队与过载
6.1.7 消除瓶颈
6.1.8 示例：I/O瓶颈
6.2 多层存储
6.2.1 内存特征
6.2.2 利用虚存管理多层存储
6.2.3 给虚存系统增加多层存储管理的功能
6.2.4 分析多层存储系统
6.2.5 存储访问的局部性与工作集
6.2.6 多层存储管理策略
6.2.7 不同策略的比较分析
6.2.8 其他页替换算法
6.2.9 多层存储管理的其他方面
6.3 调度
6.3.1 资源调度
6.3.2 调度的量度
6.3.3 调度策略
6.3.4 实例研究：调度磁盘摇臂
习题
关于第二部分
附录A：二元分类的权衡
进一步阅读推荐
问题集

术语表

章节摘录

版权页： 插图： 随着互联网的增长，一些ISP兴旺发达而另一些却没有，因此已经有一些许多合并和收购的事情。
由此导致的电子邮件服务提供商名字的脆弱性，已经创造了一个间接域名的市场。
这一市场的客户是需要稳定的电子邮箱地址的用户，例如经营私人企业或者拥有大量联系人的人们。
对于一年的费用，间接名字提供商将会注册一个新的域名，如Alice.com，并配置DNS名字服务器，以便邮箱名字Alice@Alice.com能够成为Alice@Awesome.Net的一个同义词。
那么，一旦接到ISP合并者的通知，Alice简单地要求间接名字提供商重新绑定邮箱名字Alice@Alice.com到Alice24@Awful.net，而她的联系人并不需要知道发生了什么事情。
3.3.3来自重载的更加脆弱的名字，伴随市场崩溃 美国邮政总局以层次结构分配邮递编码，称为邮政编码(zip code)，以便分发邮件时它能利用层次结构。
邮政编码有5位数字。
第一位表示10个国内地区。
新英格兰是地区0，加利福尼亚、华盛顿和俄勒冈组成了地区9。
接下来的两个数字标识地区分部。
位于马萨诸塞州波士顿的南站邮政站，是地区分部021的总部。
所有以这三位数字开头的邮政编码，使得它们的邮件在该地区分部中心进行分类。
以024开头的邮政编码标识马萨诸塞州的沃尔瑟姆地区分部。
邮政编码的最后两位数字标识特定的邮局（称为邮寄站），例如马萨诸塞州的Waban，02468。
邮政编码也有四位附加数字（称为Zip+4），用于为每位邮差依据投递顺序进行邮件分类。
尽管它们是数字的，但相邻的邮政编码并不一定会分配给相邻的站点或地区分部，所以它们真的是名字，而不是物理地址。
尽管不能作为物理地址来解释，但这些名字被重载以带有路由信息。
尽管路由是分层的，显然10个地区没有路由重要性；每件事都是分部分进行的。
据说当你走进波士顿的南站邮政站时，你会发现正在发送的邮件正在被分类成999箱，全国每个分部中心一箱。
另外，带有以021开头的邮政编码（包含南站区域）的邮件有99箱，在该区域分部内的每个邮政站一箱。
在发出的箱子中的邮件被装到袋子中，每个袋子包含发往一个地区分部的邮件。
例如，所有发往南加州地区的邮包都被装进同一辆卡车运往飞机场，在那里装入去往洛杉矶的一架飞机。
当它们到达洛杉矶从飞机中卸下时，它们又被装进不同的卡车中，开往南加州各个地区。
对于021地区的99箱邮件，也都装入袋子，每个袋子运往021地区内的一个不同邮局。
来自一个邮局并发往同一邮局的邮件仍然到地区中心进行分类，因为单个邮局没有能把邮件分成投递顺序的自动分拣机。
所有邮件都发到一个地区中心，对这一规则以前有许多例外，但现在例外的数量已经逐年减少。

编辑推荐

《世界著名计算机教材精选:计算机系统设计原理》在内容和方法上是非常独特的。
《世界著名计算机教材精选:计算机系统设计原理》深入而且广泛地介绍了构造计算机系统的主要原理和概念，这里我们所讲的计算机系统，是指广义的计算机系统，包括操作系统、客户端—服务器应用程序、数据库系统、安全的Web网站以及容错的磁盘集群等。
这些原则和抽象是不受时间影响的，不管读者是否是计算机系统专业的学生或专业人士，这些原则都具有重要价值。
这些原则和抽象经过了多代计算机系统的证明，《世界著名计算机教材精选:计算机系统设计原理》作者具有构建计算机系统的经验，并讲授了该课程数十年。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com