DISK Included

采用面向对象设计和C++的数据通讯

Data Communications Using Object-Oriented Design & C++

北京大学出版社

采用面向对象设计和C++的数据通讯

〔美〕Anil Ananthaswamy 著
柴文辛 唐 虹 译

北京大学出版社
北 京

著作权合同登记　图字：01-95-829 号

图书在版编目（CIP）数据

采用面向对象设计和C++的数据通讯/（美）安南萨斯韦姆（Ananthaswamy，A.）著；柴文辛，唐虹译．—北京：北京大学出版社，1995.12

书名原文：Data Communications Using Object-Oriented Design and C++

ISBN 7-301-02951-9

Ⅰ．采…　Ⅱ．①安…②柴…③唐…　Ⅲ．数据通讯-C 语言，面向对象语言-程序设计　Ⅳ．①TN919②TP312

书　　名：采用面向对象设计和C++的数据通讯

著作责任者：柴文辛　唐　虹

责任编辑：胡　燕

标准书号：IBSN 7-301-02951-9/TP・0274

出 版 者：北京大学出版社

地　　址：北京市海淀区中关村北京大学校内　100871

电　　话：出版部 2502015　发行部 2559712　编辑部 2502032

排 印 者：北京大学印刷厂印刷

发 行 者：北京大学出版社

经 销 者：新华书店

787×1092 毫米　16 开本　18 印张　460 千字

1996 年 1 月第一版　1996 年 1 月第一次印刷

定　　价：32 元

内 容 提 要

本书的内容包括了当今计算机界的两大热门方向——数据通讯和采用C++的面向对象设计（OOD）技术。

本书的最为可贵之处，是介绍了许多网络协议在实现过程中的细节问题，而这些细节在绝大部分的网络理论专著中是难得一见的。

贯穿本书的核心内容，是采用OOD技术完整地开发一个点到点的通讯协议——ISO 7776数据链路层协议。读者从中可以体会到一个软件从开始设计到最终实现的完整过程，从而学习到软件开发中的许多技术。

阅读本书并不需要具有OOD或数据通讯方面的背景知识，只要有初步的C++语言知识即可。本书由浅入深地介绍了数据通讯中的基本概念——层、接口、流量控制、滑动窗口协议、差错检测和校正等等。本书中使用了大量的源程序、插图和说明，介绍了如何应用OOD方法中的各种技术，如抽象、封装、模块化和继承，来开发和实现各种软件工具和网络协议。

本书中的源程序共计约5000行，稍加修改即可重新利用，具有实用价值。本书中的软件是在Unix下开发的，但对与操作系统相关的程序部分进行修改，就可以运行于其他多任务系统中。

本书确是一本不多见的论著，既可作为学习网络知识的读物，又可作为应用OOD技术的参考实例。

作 者 简 介

Anil Ananthaswamy在加利福尼亚州伯克利的TCSI公司从事软件设计工作。主要研究方向包括：网络管理系统、客户—服务器体系结构、面向对象的设计和编程、远程通讯及计算机通讯等。他获得了华盛顿大学电子工程专业的硕士学位，并在加州大学伯克利分校讲授采用面向对象设计和C++实现通讯系统的课程。

鸣　谢

仅在此向我的朋友、家人以及
所有对此书的出版作出了贡献的人们
表示感谢

在本书的写作过程中，很多人对我加以鼓励，并帮助我最终完成了这部著作。我要对下列各位表示我衷心的感谢：Sriram Srinivasan，是我在大约四年前首次提出写作本书的想法时的第一个听众，并以他的建议、灵感和渊博的知识给予了我长久的帮助。Sean Gavin，一位偶然在 Internet 网上相识的人，他阅览了全书尤其是数据通讯那一部分。我所发出的大量的 e-mail 请求都得到他的回答。我们虽然从未见过面，但是他的 Internet 地址显示他一开始是在加州的 Santa Monica，现在是在英格兰的 Berkeshire。Ralph Saavedra 对前几章提供了宝贵意见，并提醒我注意本书的语言。他现在已经很骄傲地作了父亲（当然这与本书无关）。我的编辑，McGraw-Hill 公司的 Gerry Papke 和 J. Ranade 热情地接受了本书的写作计划，在整个过程中给予我极大的支持，在本书的内容和形式上都给了我几乎完全的自由。Stephen Smith，McGraw-Hill 公司的高级编辑，对本书进行了编辑和最后的检查。我尤其要感谢 Harish Rao（VP），他对我大加鼓励，并安排 TCSI 公司对我的写作给予部分资助，这样我就可以在半天工作的同时，用半天来进行本书的写作。Ranjit Ghate，我的主管，为了便于我的时间安排而特意更改了工作日程，并且对此毫无不满（至少他没有表现出不满）。我还要感谢我的许多 TCSI 的同事，他们对我的思路（技术上和其他方面）产生了难以估量的影响，并且使我在 TCSI 的工作和学习都十分愉快。我还要感谢 TCSI 允许我使用写书时需要的资料，这些资料在通常情况下是很难得到的。

Rosu，我的一个很特别的朋友，从一开始就确信这本书是十分有价值的，并不断鼓励我坚持下去。我的许多朋友——Alka、Banny、Rao、Malini、Ramki、Praveen 和 Sundar（他对我讲述失败作家的故事以对我进行鞭策）——都用我无法用文字表达的方式对我进行了帮助。最后，我的家庭也起了不小的作用，家人对我都十分敬慕，因为我们家的一个成员正在写一本书！

前　言

尽管现在关于数据通讯的书已经非常多，既有经典之作，也有平庸无用之类，但我仍然觉得有理由再写一本关于这方面的书。

我刚从学校毕业后，曾尝试编写一个数据通讯的软件。这一段经历是促使我写此书的主要原因。尽管我读过不少优秀的教科书，但当我实际着手去实现一个著名的标准化组织推荐的协议时，我发现我以前根本没有理解和预见到开发数据通讯软件的复杂性。有很多的实际问题在任何一本教科书中都不会找到答案，这说明我还不完全具备实现一个精确定义的软件的能力。在这一段经历之后，我就计划写一本实用的和有针对性的关于数据通讯的书，重点放在那些在其他书中只是一带而过的问题上。

世上很多事情都是这样，在机遇到来之前只有等待。但在这件事情上我是幸运的，因为与此同时我正在学习面向对象的编程技术，我发现OOPS是用来解释和阐述实现协议、层和接口的具体问题的最热门的工具。所以在本书中，都是采用面向对象的方法，来研究协议层和接口的设计和实现。选用的语言是C++，主要因为这是我最熟悉的语言。

本书是围绕着ISO7776标准——一个数据链路协议的工业标准——的实现来展开讨论的。即使是只实现一个完整的层，也需要我们去研究层和接口这些基本概念。书中作为例子的源代码部分尽量与协议标准相一致。代码部分不能保证完全没有错误，欢迎广大读者能发现其中残留的错误并提出改正办法。

在此有必要强调一下本书没有详加讨论的内容。本书不是关于数据通讯、面向对象的设计和编程方面的综合性论著。可以作为高年级本科生或研究生的计算机网络和数据通讯课程的辅助读物，提供一个这类问题的实际例子。本书的读者需要具有C++的实际经验，并熟悉Unix操作系统。

随书附带一张软盘，其中存有书中所用到的全部源代码。盘中存放的是Unix的tar格式文件，用“tar xvf〈devicename〉”可以将此文件还原。实际上在书中出现的大多是程序的片段，有的还需要对源码进行一些修改。如果书中的代码和盘中的内容出现不一致时，可以认为盘上的内容是最终的正确版本。这些源代码也可以用ftp命令从ftp.uu.net上得到，在/published/mcgraw-hill/datacomm-ood/目录下存放着全部源代码的tar格式压缩文件datacomm.tar.Z（本书中所出现的源程序和程序片断可以免费自由使用，但请注明出处）。

我尽最大努力以使本书更加准确，但错误之处在所难免。书中出现的所有错误都由我负全部责任，并欢迎各位读者通过McGraw-Hill公司向我指出书中的各种错误（概念上的、可行性上的和编程方面的），任何对本书加以改进的建议都将受到欢迎。

Anil Ananthaswamy

目　录

第一部分　概　念

第二部分 程序框架

第三部分 协议的实现

第一部分　概　　念

本书中的前五章将主要讨论实现一个协议层过程中的基本概念。协议、层和接口将在第二章中介绍。面向对象的设计和编程将在第三章中介绍。第四章中将仔细研究协议的各组成部分，以及将要实现的协议中各个过程的基本原理。第五章提供一个高层的设计，并对整个实现过程加以概述。

第一章 概 述

1.1 本书的目的

本书主要有两个目的：首先，使读者对具体实现一个数据通讯协议、层和接口过程中的概念和细节有一个明确的认识。其次，了解采用面向对象的设计和实现时，进行开发的过程。本书将通过一个采用面向对象的设计和实现的数据通讯协议、它的层、以及它的接口，来详细阐述这两方面的内容。

协议定义了两个实体相互通讯时的方式。计算机通讯包括了不同层次上进行通讯的许多协议（比如从传输位、字节，到传输文件）。为了降低各个协议的复杂程度，各协议在逻辑上和物理上都组织成一系列叫作层的实体，每个层只提供某种特定的服务（比如一个层负责传输位，而另一个层负责传输文件）。层与层之间通过精确定义的接口进行交互。

1.2 具体实现者的看法

写作这一小节的目的是：尽管从理论上了解协议、层和接口这些概念是必需的，但这些对于理解它们是如何相互配合进行工作还是不够的。这里我们通过具体实现一个 ISD7776 标准所定义的数据链路层协议，来看一下具体实现者的观点。为了能够进行数据传输，我们用 Unix 的 IPC 机制实现了一个物理层。同时我们还要实现一个简单的驱动层，以便与数据链路层接口。驱动层也作为用户接口，允许我们通过它完成调用数据链路层的服务。有必要强调的是，虽然我们只实现了一个数据链路层，但是基本思想对于实现其他层也是适用的。我们在实现过程中所要用到的基本概念包括：

- 分层的原理
- 层间的接口
- 协议的构成要素
- 流量控制机制
- 差错检测和差错校正

要实现这样的一个任务需要大量的相关工具和实用程序，我们将对此详加讨论：

- 跟踪和调试工具
- 表
- 异步编程原理
- 有限状态机

最后，我们简要介绍一下协议的一致性检查这一概念。我们的协议可以通过用户界面进行测试，输入的数据中应包含有错误的条件，这样协议的性能就可以在测试中表现出来。

如果你不熟悉数据通讯中的术语，像 OSI、ISO、数据链路层、物理层等等，请不要泄气，我们将在后面陆续解释这些术语。

1.3 面向对象的设计和编程

本书的另一个特点就是采用面向对象的设计和编程方法来完成我们的任务。OOD 和 OOP 是独立的研究领域，在此，我们只想通过解决一个实际问题，即采用面向对象的方法，来实现一个协议、层和接口，使读者对 OOP 和 OOD 有所了解。在设计方法上我们没有任何限制，主要是想使读者对设计过程有一个感性的认识。我们要解决的问题具有足够的难度，这正好能充分发挥面向对象方法的能力来加以解决。在设计过程中需要不断强调的 OOD 中的基本概念有：

- 抽象
- 封装
- 模块化
- 继承

最后，代码的可重用性也是 OOD 的优点之一，具体的实现方法将在后面介绍。

1.4 过　　程

在写作本书时，我是出于这样一种考虑，那就是要想理解像数据通讯这样一个专业问题，最好的办法就是自己动手去实现。我们只讨论协议实现过程中的某些特定内容，通过对这些局部的了解，就能加深对整体的复杂性的认识。

在本书中正文的顺序安排如下：首先提出问题，分析问题，设计一个解决的方法，然后实现这个方法。只有当要解决的问题被精确地定义后，这个方法才会有实际意义。在设计过程中的每一步，我们都把问题尽量阐述清楚，然后才去进行方法的设计，最后去具体实现。书中的程序是在 Unix 平台上使用 C++ 开发的，具有足够的模块化程度，稍加修改就可以移植到其他操作系统中。

1.5 源　代　码

书中的 C++ 源代码是本书的一个重要组成部分。许多优秀的软件书籍，不论是专著还是教科书，都很好地使用了源代码。源代码能够有效地说明实现的方法和基本原理。在书中有多处使用了源代码而不是文字来阐述一个问题。书中大量使用的是源代码的一些片断，而不是完整的函数、类和程序。如果觉得这样不习惯，随书提供的软盘中包含有书中所有的源代码，将这些源代码打印出来，可以成为本书的补充，随时参考。

1.6 正文的组织

本书是围绕着一个协议层的实现而组织的，全书共分三部分。这样划分是因为我们考虑到，有的读者可能只对书中的某些部分感兴趣，这样他们就能跳过不必要的内容，直接从他们需要的地方开始。

1.6.1 第一部分——基本概念和高层设计

第一部分主要介绍基本概念。

· 第一章是概述。

· 第二章则重点介绍数据通讯软件中分层的原理，特别是OSI的层次结构，详细介绍层、服务和接口。同时讨论协议的构成元素。

· 第三章详细讨论面向对象的设计和编程的原理。

· 第四章重点分析ISO 7776协议，介绍协议的工作原理。协议的各个部分都将仔细讨论。

· 第五章详细讨论了要解决的问题，并从高层上提出了解决的方法。

1.6.2 第二部分——实现的准备工作

第二部分着手开发一个实用程序库，以作为后面正式实现协议时的开发环境。实际上，这里所设计的这些实用程序在其他类似的项目中是可以重新利用的。这一部分的主要目的是使读者了解问题的实质，从而找出解决的办法。各章的安排如下：

· 第六章中设计了一套通用实用工具，包括跟踪/调试、表、队列和堆栈，以及用于在层间进行数据交换的数据包类。

· 第七章分析异步系统的特点，开发一系列用于异步编程的类，包括调度、事件处理和定时器。

· 第八章介绍有限状态机（FSM）的概念。重点放在用面向对象的方法实现FSM上，实现的方法是传统的表驱动FSM法。利用FSM的概念，在Unix的IPC上开发一个物理层。作为开发平台的完善工作，再实现驱动层的主要元素。

1.6.3 第三部分——协议的实现

第三部分着重介绍实际协议和层的具体实现。各章的安排如下：

· 第九章开始进行具体设计，定义数据链路层FSM的状态和事件，定义实现FSM所需要的状态机类。数据链路层所需要的其他类也在此定义。

· 第十章介绍数据链路层中链路管理的实现。

· 第十一章介绍流量控制和滑动窗口协议的概念和实现。

· 第十二章介绍协议中的差错检测和差错校正部分。

· 第十三章完成所有剩下的内容。用户接口、协议的一致性检查、以及测试方法。

· 附录A介绍本书中所使用的帧检验序列的计算方法。

· 附录B介绍本书中关于有限状态机的约定。

· 附录C中列出了一些有用的头文件（其中包括常量定义和外部声明）。

· 附录D介绍了数据通讯领域的趋势。简单介绍了LAPD和帧转发技术。

1.7 小　结

本书通过一个具体实现的例子，涉及到了数据通讯和面向对象的设计中的一些重要概念，主要内容如下：

- 层和接口
- ISO 7776 数据链路层协议
- 面向对象的设计和实现的基本原理
- 流量控制和滑动窗口协议
- 差错检测和差错校正
- 异步编程、调度和事件处理的基本原理
- 有限状态机的实现
- 物理层—数据链路层以及数据链路层—网络层接口
- 可重用的面向对象库，比如跟踪/调试、表、栈和队列

第二章　协议、层和接口

2.1　简　　介

在过去，能说出OSI参考模型中七个层的名称，就可以在数据通讯公司中找到一份工作。现在不同了，要找一份数据通讯类的工作需要有很多的背景知识，但是不管你信不信，在面谈时仍会问到这七个层的名字。尽管本书中只讨论实现某个特定的协议、层及其接口，为保持完整性我们还是会讨论OSI参考模型。我们首先讨论协议的概念，在数据通讯中分层的必要性，以及由于分层而带来的接口和服务。最后，讨论具体实例：网络层—数据链路层接口，以及数据链路层—物理层接口。

2.2　协　　议

让我们先来看两个人在电话上交谈的情况。为了能够进行有效的交流，双方都必须遵守某些规则。比如，两个人都要使用对方能听懂的语言，这里的语言既指词汇又包括语法。同样，这两个人要轮流讲话，都有机会说和听。在真正开始交谈之前，还要按照某种过程去操作，以便能打通这个电话。所有的这些规则加在一起，就是进行电话交谈的协议。

同样地，计算机世界中的两个实体需要通讯时，也需要建立一个协议，来定义双方交换信息的方式。

Holzmann[1]把下列各项认为是理解一个协议的基本元素：

- 协议所提供的服务
- 协议所运行的环境
- 协议信息所使用的词汇
- 每条信息使用规定词汇的编码格式
- 保持信息交换稳定的操作规则

当我们后面讨论要实现的协议时，还会提到这些基本元素，并按照这样的线索进行学习。

2.3　分层的必要性

由于计算机和计算机通讯技术的飞速发展，再加上各公司所提供的各种不同的服务，协议的数量和类型都将变得难以控制。而分层就是解决这一难题的办法。

图2.1所示是整体实现文件传输的例子。考虑两个实体A和B，通过一条物理介质交换信息，允许A和B这两个用户在各自的系统之间传输文件。这样的一个文件传输系统应该提供下列功能：

- 一个传输协议，提供ASCII数据的传输能力。
- 一个用户接口。

·与硬件相关的软件来与物理介质（同轴电缆）进行接口。

整个的程序是一个单元，没有按功能分离成独立的模块。从编程(C/C++)的观点来看，就是：

·一个单独的 main （） 函数。

·共享全程变量。

·代码高度耦合。

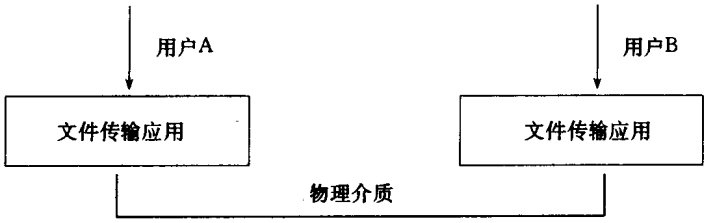

图 2.1 整体实现

如果系统环境发生变化，这种设计就会产生问题。请看下列情况：

·如果由于数据传输技术的进步，传输介质从同轴电缆改为光纤，整个的程序都要修改，以适应新的介质。

·如果要求传送二进制数据，而现在的文件传输协议只能传递 ASCII 数据，则协议必须进行修改以适应二进制数据的要求。这同样意谓着要对这一软/硬件一体的单元进行修改，才能加入新的功能。

考虑到当今科学技术的发展速度，要想维护这样一个软/硬件一体化的系统不断更新，这简直是痴人说梦。

正是在这种情况下引入了分层的机制。图 2.2 显示了把原来的作为一个整体的程序按照功能进行组织的情况，从而把高度复杂的整体程序变成了易于管理的几个部分。通过把整个数据通讯系统划分成一个硬件相关层、一个文件传输层和一个用户接口层。我们现在能够把程序按照功能进行隔离，这样在某一层中的变化就不会影响到其他层。

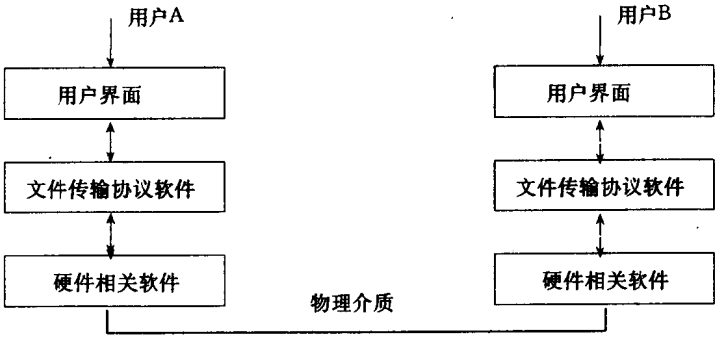

图 2.2 分层

比如，如果物理的传输介质要进行更新，只有硬件相关层需要相应进行修改，而上面的两层可以保持不变。同样，如果文件传输协议发生改变，所有的修改也都局限在实现协议的那一层中。从编程的角度来看，分层使对软硬件的维护、升级和修改都更加容易。