

基于和欣嵌入式操作系统的构件化TCP/IP协议栈

胡代前, 陈榕

(同济大学 计算机科学与工程系, 上海 200092)

摘要: “和欣”嵌入式操作系统是基于CAR构件技术^[1]、支持构件化应用的操作系统, 是国家 863 支持的TD-SCDMA的操作系统标准。本文首先分析了当前嵌入式系统的发展对网络支持的需要, 然后介绍了和欣操作系统以及CAR构件技术, 最后在介绍了在和欣上TCPIP协议栈的特点及实现。

关键词: 和欣嵌入式操作系统; CAR 构件技术; TCP/IP 协议栈;

Hu Daiqian, Chen Rong

(Department of Computer Science and Engineering, Tongji University, 200092, China)

Abstract: Elastos is an embedded operating system based on CAR component technology, and the standard TD-SCDMA operating system supported by the "863" Program. In this paper, it shows the needs of the network support for the current development of emdedded system first, and then Elastos embedded operating system and CAR component technology are given. At last, it shows the specialty and implementation of the TCPIP stack on the Elastos.

Key words: Elastos embedded operating system; CAR; TCP/IP stack;

0 引言

嵌入式Internet是近几年随着嵌入式系统的广泛应用和计算机网络技术的发展而发展起来的一项新兴概念和技术。嵌入式系统以应用为中心、以计算机技术为基础、软件硬件可裁剪, 赢得了巨大的市场, 在应用数量上远远超过了各种通用计算机。随着 Internet的发展, 各种设备都产生了连接性的需求, 从冰箱到电表, 似乎所有电器都需要连入互联网。通过为现有嵌入式系统增加因特网接入能力来扩展其功能, 以Internet为介质实现信息交互的过程, 这就产生了嵌入式Internet 技术。要实现嵌入式设备的网络化, 需要实现TCPIP网络协议栈, 但由于Internet上面的各种通信协议对于计算机存储器、运算速度等的要求比较高, 对嵌入式系统产生了很大的挑战。

目前有两个关键的因素影响网络协议栈的开发, 一是性能和效率, 二是开发调试方便。传统的操作系统一般只能做到其中的一个方面。例如在Linux, FreeBSD等传统操作系统中, 网络协议栈是作为系统内核的一部分实现的, 通常具有较高的通信效率, 但是不利于用户添加实现新的用户协议, 以及根据自己的特殊需要对协议的实现细节进行调整, 也不利于协议代码的维护、调试和开发。这个缺点在嵌入式领域表现得尤为突出。

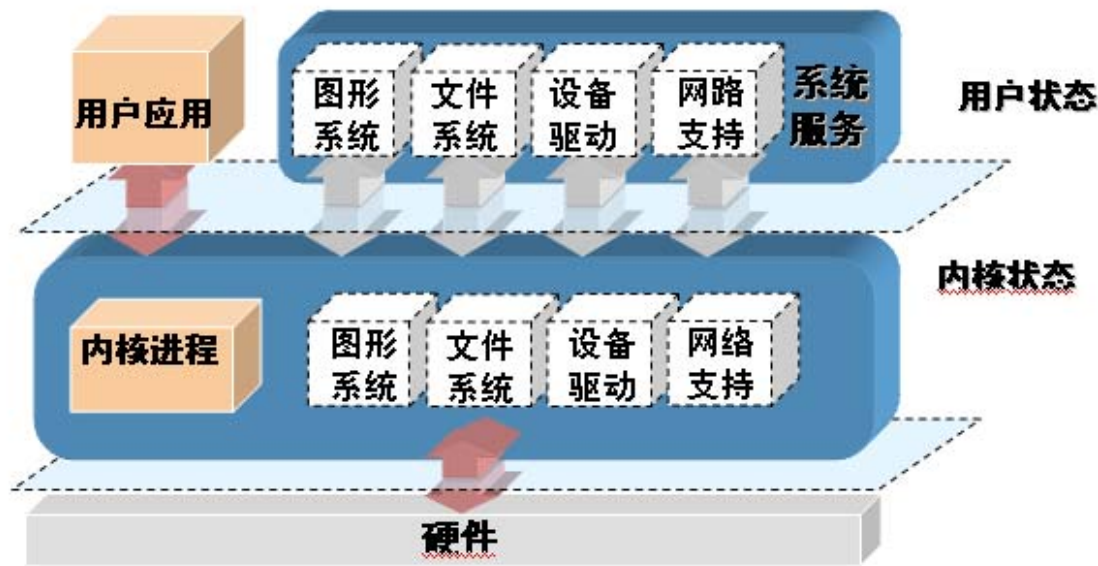
随着嵌入式设备的广泛普及和嵌入式设备对网络接入的需求, 快速简单地开发应用于嵌入式设备的网络协议栈显得非常必要。但是由于嵌入式设备的多样性和通信需

求的多样性，嵌入式领域的网络协议栈的复用率非常低。该文章将构件技术引入网络协议栈的开发，并在国产的和欣操作系统上实现了一个构件化的网络协议栈，很好地解决了开发与效率的问题。

1 关于和欣嵌入式操作系统和 CAR 构件技术

1. 1 和欣嵌入式操作系统

和欣操作系统基于微内核，服务动态加载的策略，内核只包括硬件驱动（串口，USB，以太网卡，无线通信模块），内存管理等系统基本服务（图一），之上是构件运行环境，而TCP/IP协议栈则是完全由构件实现，运行在构件平台之上的一个服务构件。底层网卡，串口等设备得到数据之后将数据交给上层的服务构件来处理。



图一

和欣嵌入式操作系统是一个基于构件化软件模型的系统，构件化软件设计思想贯穿了整个系统的设计与实现中，系统实现本身就是构件模式。除内核中最底层的控制部分外，所有系统功能都是以构件接口的形式提供。另外，操作系统对构件化软件模型提供了必要的运行环境，来源不同的构件可以在该环境上实现互操作。系统提供了构件自动寻址/自动加载机制，用户不必知道调用的构件程序是本地的还是来自于网上，也就是说，构件运行环境可以对用户透明。构件化系统的实现，使得操作系统本身具有高度的灵活性和扩展性。和欣采用的构件技术是 CAR 构件技术。

1. 2 CAR 构件技术

CAR 是 Component Assembly Runtime 的缩写，CAR 构件技术定义了一套网络编程时代的构件编程模型和编程规范，它规定了一组构件间相互调用的标准，使得

二进制构件能够自描述，能够在运行时动态链接。CAR 构件技术就是在总结面向对象编程、面向构件编程技术的发展历史和经验，为更好地支持面向以 Web Service（WEB 服务）为代表的下一代网络应用软件开发而发明的。

CAR构件技术通过二进制的封装以及动态链接技术解决软件的动态升级和软件的动态替换问题。面向构件技术对一组类的组合进行封装，并代表完成一个或多个功能的特定服务，同时为用户提供多个接口。整个构件隐藏了具体的实现，只用接口提供服务。这样，在不同层次上，构件均可以将底层多个逻辑组合成高层次上的粒度更大的新构件。构件之间通过约定的接口进行数据交换和信息传递，构件的位置是相互透明的，可以在同一个用户进程空间，也可以在不同的用户进程空间，甚至在不同的机器上，而且不同的构件可以用不同的语言编写，只要它们符合事先约定的构件规范。

2 和欣操作系统上的 TCP/IP 协议栈的特点与实现

2. 1 和欣嵌入式操作系统上 TCP/IP 协议栈的特点

和欣是嵌入式操作系统，而由于嵌入式系统在硬件资源的占有量以及处理器的处理能力等方面的限制,使得TCP/IP 协议的嵌入式实现与其在通用操作系统中的实现有很大不同。首先,由于嵌入式处理器的时钟频率低，地址、数据总线窄,导致嵌入式系统对一个通用IP 包的处理要花费更多的处理机时间,从而影响其他任务的执行。因此就需要对原本庞大复杂的TCP/IP 协议栈进行裁剪,使之具有简单、高效的特点。同时在设计嵌入式TCP/IP 协议栈时要合理的控制中断处理程序的大小,使得中断处理程序的运行时间尽可能的缩短;同时要把那些无实时性要求和费时的的工作移到主程序中去执行,从而保证协议的正确执行。具体来说就是把TCP/IP 协议的处理工作放到主程序中,对网络接口的控制芯片采用循环查询的方式,在其他中断任务的执行间隙来处理TCP/IP 协议,以牺牲系统的响应时间来换取系统的可靠性。再者由于嵌入式系统的存储空间有限,一般情况下只能开辟几K字节的缓存区来接收数据包,并且其存储空间是固定的,而不是动态分配的。如此一来在遇到大数据包的时候就会出现缓冲区溢出的错误。不过幸运的是在大多数情况下,嵌入式系统接收的数据包的类型仅为系统状态配置包、控制动作命令包、通信控制包等,包容量较小;而那些实时数据或网页向网络发送的数据包就可能比较长,但却可以通过设计格式来加以控制。

和欣操作系统是基于微内核的操作系统，因此其上的TCP/IP协议栈既可以运行在用户态，又可以运行在内核态，提供一个通用的接口和一个方便的框架方便用户开发添加适合自己领域或特殊需要的网络协议栈。因为和欣操作系统提供了高效的进程上下文切换功能和高效的进程间通信，所以整个网络协议栈作为一个用户态进程提供服务。使用前先创建网络构件实例，然后向操作系统注册服务。此后用户就可以使用网络编程接口了。如果用户对网络协议栈有更高的效率要求，可以将该协议栈加载到内核态。

2.2 和欣嵌入式操作系统 TCP/IP 的实现

根据和欣嵌入式TCP/IP协议栈的特点,在实现嵌入式TCP/IP协议栈的时候只需要用到下述四个基本协议: TCP 协议、ICMP 协议、IP 协议、ARP 协议。其余的像PPP那样的链路层协议就需要在嵌入式TCP/IP协议栈下面的设备驱动程序中实现。而像HTTP、FTP这样的应用层协议则要在嵌入式TCP/IP协议栈上面的应用层中实现。

2.2.1 地址转化协议- ARP 协议

ARP 协议为IP地址和以太网MAC地址建立一个映射表。这个协议对于以太网的TCP/IP协议的操作是必不可少的。为了节省空间,一个IP地址的ARP请求覆盖即将发送的数据包。该ARP请求就是因为要发送这个包而发送的。这种方法是在假设上层会重发被覆盖的数据的情况下采用的。每隔十秒钟,表就会刷新一次,旧的记录会被丢弃。在嵌入式TCP/IP协议栈中,记录的有效时间为1秒。

2.2.2 网际协议- IP 协议

在嵌入式TCP/IP中,IP协议执行两项功能:一项功能是封装来自上层的TCP数据包或ICMP数据包,计算IP数据包的检验和;另一项功能是解析IP缓存中的数据包,检查数据包的协议字段,以判断该数据包应该交给ICMP协议还是TCP协议处理。基于嵌入式WEB服务器与客户端通信的信息量很小,数据包的大小不会超过链路层的MTU(最大传输单元)的事实,嵌入式IP没有引入数据包分片和重组的功能。标准的IP协议有选路的功能,可以选择较短路径传送数据包,在嵌入式IP中,为了减少代码的规模,没有引入选路功能。

2.2.3 网际控制报文协议- ICMP 协议

在嵌入式TCP/IP协议栈中,只处理一种ICMP消息类型:ICMP回送消息。ICMP回复消息经常被使用,那就是调用ping程序来测试对方主机是否在线。在嵌入式TCP/IP协议栈中,ICMP回送消息是用一种十分简单的方式实现的。将ICMP类型的字段由“echo”类型改变成“echo reply”类型。改变IP首部的IP地址,将数据包送回到发送者那里。

2.2.4 传输控制协议- TCP 协议

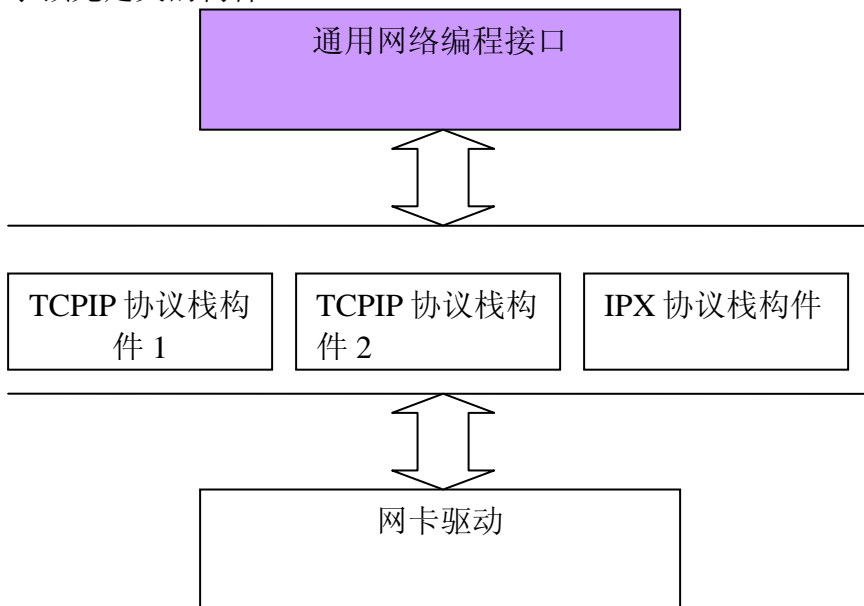
TCP协议是面向连接的、端到端的可靠通讯协议。它采用了许多机制来保证传输可靠性,应用于嵌入式系统显得过于复杂。在嵌入式TCP/IP协议栈中,为了减少存储空间的占用,没有引入发送和接收数据的可靠窗口机制。作为嵌入式WEB服务器,一般处于被动服务的状态,所以可以将标准TCP有限状态机的主动创建连接的SYN SENT状态、主动关闭连接的FIN WAIT1、FIN WAIT2、CLOSING、TIME WAIT状态裁剪。在设计中,还去掉了LISTENING状态,让它在CLOSED状态的情况下,就处于侦听状态,来监听客户端的连接请求,这样就避免了主动或被动打开和关闭的操作,使嵌入式TCP/IP协议栈更精简。嵌入式TCP/IP不缓存到达的TCP段,应用程序必须立即处理它。当然,这并不阻止应用层它自己缓存数据。每次连接在发送数据时,无法同时处理多于一个的TCP段。TCP协议的实现原理比较复杂,在嵌入式TCP/IP协议中,实现TCP协议是一个最关键的问题。

2.2.5 构件化的协议栈

采用构件化方法实现网络协议栈的关键之处在于以下三点：

- (1) 如何划分构件的粒度；
- (2) 如何设计构件接口；
- (3) 提高网络通信的效率。

第一点，需要根据用户的需要决定，如果用户是开发一个新的协议栈，可以采用大粒度的构件划分方式（如图2所示），即把整个网络协议栈作为一个大的构件，继承预先定义的构件。



图二 构件化协议栈框图

第二点，要实现第一点所提到的构件的自由替换，首先必须设计良好的构件接口。所有的构件实现应该继承该接口。设计接口的原则是尽量保持接口的通用性和简洁性。当有新的功能添加时，可以通过继承实现新的接口方便地加入原有的框架，并不影响原来程序的正常使用。从而实现软件的无缝升级。

第三点，一个高效的通信协议需要满足以下三个方面：①上下文切换和定时器的轻型完成；②多个协议具有统一的使用界面；③在网络设备，核心层和用户层之间有高效的缓冲机制，避免不必要的拷贝。

如果操作系统提供了快捷的上下文切换功能以及高效的进程间通信，那么整个网络协议栈可以作为一个用户态的进程对用户进程提供服务。如果进程上下文切换比较复杂，则整个网络协议栈可以作为动态链接库链入用户进程地址空间，但是这时需要对多进程共享的数据加锁进行同步访问控制。另外一种选择是将网络协议栈装入内核空间，这样也可以避免频繁的进程上下文切换带来的系统开销。

因为采用了通用的SOCKET编程接口作为最终提供给用户使用的接口，这样就使多个协议具有统一的使用界面。同时，在采用细粒度构建的网络协议栈中，各层协议之间的接口也是精心构造的统一的接口。为了避免大量数据的拷贝，需要设计良好的数据结构，同时可以采取“共享内存”的方法。在网络协议各层次之间尽量以指针的方式传递数据。

构件化的网络协议栈具有以下优点：

(1) 针对不同网络应用的需求, 能最大限度地利用领域相关知识进行调整, 提高了性能。

(2) 底层构件可以在不影响顶层使用的情况下进行修改、

(3) 方便用户开发调试新的协议, 方便用户添加新的协议

3 总结与展望

该文采用上海科泰公司的CAR构件技术在和欣操作系统上实现了一个构件化的网络协议栈。该网络协议构件既可以加载在内核态, 也可以加载在用户态, 方便开发调试, 有利于开发新的网络协议栈。其他开发者采用构件的方法开发出了高效或者针对某特定领域的网络协议栈, 可以方便地替换和欣操作系统上原有的网络协议栈。使用构件化的方法开发网络协议栈也可以推广到其他支持构件的操作系统上。

参考文献

[1] Wright G R, Stevens W R. TCP/IP Illustrated Vol 2: The Implementation. Boston MA USA, Addison Wesley, 1995

[2] Stevens W R. TCP/IP Illustrated Vol 1: The Protocol. Boston MA, Addison Wesley, 1994

[3] Karn P. KA9Q TCP/IP Source Code.
<http://www.ka9q.net/code/ka9qnos/>, 1993

[3] (美) R. Braden, D. Borman, and C. Partridge. Computing the internet checksum. RFC 1071, Internet Engineering Task Force, September 1988

[4] (美) W. Richard Stevens. “TCP/ IP 详解. 卷一: 协议[M]. 范建华, 等, 译. 北京: 机械工业出版社, 2000

[5] (美) Gary R. Wright, W. Richard Stevens. “TCP/ IP 详解卷二: 实现[M]. 陆学莹, 译. 北京: 机械工业出版社, 2000

[6] (美) McCombie B. Embedded web servers now and in the future. Real - Time Magazine, 1998, 1: 82~83

[7] (美) J. Postel. Internet protocol. RFC 791, Internet Engineering Task Force, September 1981

[8] (美) Comer, D. E. . 用TCP/ IP 进行网际互联第一卷: 原理、协议和结构(第四版) [M]. 林瑶, 等, 译. 北京: 电子工业出版社, 2001

本文受国家 863 计划“软件重大专项”支持(课题名称: 基于中间件技术的因特网嵌入式操作系统及跨操作系统中间件运行平台, 课题编号: 2001AA113400, 所属专题: 计算机软件, 所属领域: 信息技术领域)
胡代前(1981-), 男, 重庆人, 硕士研究生, 主要研究方向: 系统软件支撑技术; 陈榕(1952-), 男, 教授。