

文章编号:1001-9081(2007)09-2310-02

## ElAura 跨操作系统构件运行平台设计与实现

陆 刚, 陈 榕

(同济大学 基础软件工程中心, 上海 200092)

(lugang@elastos.com.cn)

**摘 要:**为了在多种操作系统上提供虚拟统一的 CAR 构件运行平台,分析了和欣操作系统 (Elastos) 及 CAR 构件技术的特点,提出了 ElAura 构件运行平台的分层模型设计,研究了关键技术的实现方法。该平台的成功实施,使得 CAR 构件在任意操作系统或软件平台上能够独立部署、正常运行。

**关键词:**二进制兼容;构件;中间件

**中图分类号:** TP311 **文献标志码:** A

### Design and implementation of ElAura component runtime platform

LU Gang, CHEN Rong

(System Software Engineering Centre, Tongji University, Shanghai 200092, China)

**Abstract:** In order to implement a virtual and universal Component Assembly Runtime (CAR) platform on multi operating systems, the characteristics of the Elastos operating system and CAR technology were analysed, and the hierarchical model of ElAura component runtime platform was presented. Then the implementation of key techniques was described in detail. Based on ElAura, a binary component could be distributed independently and run properly on any operation system or software platform.

**Key words:** binary compatibility; component; middleware

## 0 引言

CAR 技术作为新一代构件技术之一,必然要实现网络环境下跨平台部署、分布式运行的功能。开发 ElAura 的主要目的是使网络环境下的 CAR 构件能够跨操作系统运行。ElAura 构件运行平台的实现将促进 CAR 构件标准的推广,同时原本只能在和欣系统上运行的应用程序和网络 CAR 构件可以直接在其他宿主操作系统上运行,不仅扩展了宿主系统的功能,也促进了宿主系统和和欣双方的应用与发展。

在 Windows、Linux、Windows CE 等其他操作系统上,ElAura 构件运行平台屏蔽了底层传统操作系统的具体特征,实现了一个统一的运行环境。在和欣上开发的应用程序,可以不经修改、不损失太多效率、以相同的二进制代码形式,运行于传统操作系统之上。缩小了应用软件平台间移植的周期,提高了软件复用性。

## 1 和欣及 ElAura 构件运行平台

和欣是基于微内核,具有多进程、多线程、抢占式、基于线程的多优先级任务调度等特性的 32 位嵌入式操作系统<sup>[1]</sup>。和欣完全面向构件技术,提供的功能模块全部基于 CAR 构件技术,因此是可拆卸的构件,应用系统可以按照需要剪裁组装,或在运行时动态加载必要的构件。和欣操作系统体积小,速度快,适合网络时代的绝大部分嵌入式信息设备。

在和欣操作系统中,ElAura 构件运行平台与和欣灵活内核共同构成了完整的操作系统。ElAura 构件运行平台提供了

一套符合 CAR 规范的系统服务构件及支持构件相关编程的 API 函数,实现并支持系统构件及用户构件相互调用的机制,为 CAR 构件提供了编程运行环境。ElAura 构件运行平台在不同操作系统上有不同的实现,符合 CAR 编程规范的应用程序通过该平台实现二进制级跨操作系统平台兼容。ElAura 构件运行平台直接运行二进制构件,而不是像 Java 和 .NET 那样通过虚拟机在运行程序时解释执行中间代码<sup>[2]</sup>。因此,与其他面向构件编程的系统相比,具有资源消耗小,运行效率高的优点。

## 2 ElAura 体系结构设计

ElAura 构件运行平台采用了分层的模型。平台适配层、和欣模拟层、服务层这三部分共同组成了 ElAura 构件运行平台。在实现了平台适配层之后,和欣模拟层就不需要了解底层操作系统的细节,不需要根据底层操作系统的不同而重新实现。在我们的实现中,和欣模拟层在各个操作系统上是二进制兼容的。服务层要同时使用和欣模拟层的功能(如创建线程、同步对象等)和底层操作系统的功能(如读写文件、网络通讯等),所以它同时依赖于这两层。和欣的服务是基于 CAR 构件的,所以在不同的操作系统上,只需将服务的 CAR 构件实现直接替换即可。

### 2.1 分层模型

ElAura 构件运行平台能够顺利完成的关键,就是采用了分层的设计。上一层的模块只依赖于下层的模块,没有反向依赖,这使得实现人员的知识可以局部细化,提高开发效率,

收稿日期:2007-03-15;修回日期:2007-06-08。 基金项目:国家 863 计划项目(2001AA113400)。

作者简介:陆刚(1982-),男,浙江绍兴人,硕士研究生,主要研究方向:嵌入式系统、构件技术; 陈榕(1952-),男,北京人,教授,博士生导师,主要研究方向:嵌入式系统、构件技术。

并且分工明确。负责平台适配层,只需具有目标操作系统的知识,而负责和欣模拟层,只需和欣操作系统的知识,相互之间的耦合度非常小。

图 1 显示 ElAura 构件运行平台的层次模型。图中的箭头表明了层次之间的依赖关系。和欣应用程序,是由用户或者其他开发商开发的 CAR 构件,是二进制兼容运行的目标。ElAura 就是为这些应用程序提供运行的环境。对于应用程序来说,它们无法分辨自己是运行在真正的和欣操作系统环境中,还是运行在其他操作系统上。应用程序只依赖于和欣为它们提供的应用程序编程接口、内核对象接口和一些系统的服务(如文件系统、TCP/IP 网络、设备驱动等)。

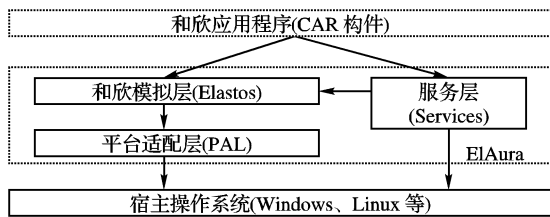


图 1 ElAura 分层模型

### 2.2 平台适配层

平台适配层屏蔽了底层的宿主操作系统,利用宿主系统 API 来实现和欣模拟层所需的原语。这些面向过程的原语与 Win32 API 非常类似,因此在语义上和欣操作系统的语义要简单。可以用平台适配层的语义来构造更为复杂的和欣操作系统的语义。如图 2 所示, PAL 层的实现要结合宿主操作系统的知识,在不同的操作系统上有不同的实现方案。

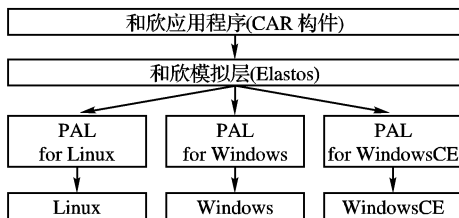


图 2 平台适配层原理

### 2.3 和欣模拟层

和欣模拟层向用户提供了一套支持 CAR 规范的系统构件及支持构件相关编程的 API 函数。和欣操作系统的应用程序编程接口与 Windows 等操作系统的应用程序编程接口的不同之处,就是和欣操作系统采用了面向构件的方式,而不是 Windows 等操作系统面向过程的方式。和欣模拟层模拟了和欣操作系统的语义,和欣操作系统的语义实际上就一组逻辑的组合,比如进程的逻辑、线程的逻辑、事件的逻辑、互斥锁的逻辑等。只要提供合适的平台适配层原语,就能够完全构造出和欣操作系统的语义。

### 2.4 服务层

因为和欣操作系统的系统服务和设备驱动采用了基于 CAR 构件的服务模型,所以可以方便地设计出服务层。面向构件编程模型的优点之一就是用户通过构件的接口获取构件的功能,用户并不关心构件的实现细节,而构件本身就像零件一样,可以替换和更新<sup>[3]</sup>。和欣操作系统将驱动程序和文件、网络等服务都设计成构件的方式,因此,在其他操作系统中实现服务层时,只需用宿主系统的功能重新实现驱动和服务构件并将其替换即可。

## 3 ElAura 构件运行平台关键技术实现

### 3.1 PE 格式文件加载

对应用程序的加载和卸载是 ElAura 的首要任务。操作系统本身的应用程序加载器只能识别自己支持的可执行文件格式。Elastos 和 Windows 操作系统的可执行文件格式是 PE 格式,而 Linux 的可执行文件是 ELF 格式<sup>[4]</sup>。因此对于像 Linux 这种不支持 PE 格式的操作系统,ElAura 必须提供 PE 文件加载器。

PE 加载器本身是目标操作系统支持的一个应用程序,因此可以在目标操作系统上运行。PE 加载器会把目标操作系统不支持的可执行文件加载到自己的地址空间内,并跳转执行,从而达到加载的目的。图 3 是 PE 加载器工作原理图。

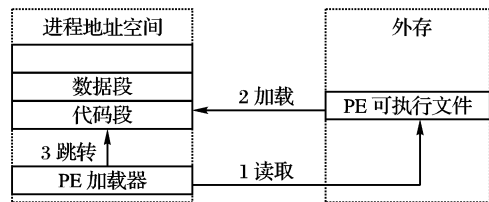


图 3 PE 加载器原理

在加载开始前,在目标操作系统上运行 PE 加载器,因为 PE 加载器本身是目标操作系统所支持的可执行文件格式,因此目标操作系统会创建一个新的进程,然后把 PE 加载器本身加载到新的进程地址空间内。PE 加载器会从外存和文件系统中读取要加载的 PE 可执行文件,并解析文件格式。然后,PE 加载器会在自己的地址空间内开辟一段新的内存,把可执行文件的代码段、数据段加载到内存中。如果代码需要重定位操作,PE 加载器还帮助代码段进行重定位,如果可执行文件有依赖关系,PE 加载器会把它所依赖的部分也一起加载起来。经过这些步骤之后,可执行文件就已经在进程的地址空间中以可运行的状态存在。最后,PE 加载器会跳转到新加载到内存中的代码段开始执行。

通过 PE 加载器,可以解决目标操作系统本身的加载器不能加载和欣应用的可执行文件的问题,让和欣程序在目标操作系统上运行成为可能。应用程序运行结束卸载的时候,依然是目标操作系统自动回收该进程占据的所有资源,无须进行更多的干预。

### 3.2 内核对象模拟

与传统操作系统不同,和欣操作系统内核的各个功能模块按照 CAR 构件规范建立,都是 CAR 构件对象。和欣内核向和欣应用程序提供的大部分系统服务是内核对象的接口,只有少量系统服务是直接以 API 函数形式提供。

内核对象主要负责进程、线程管理,同步、互斥、消息队列、管道等。为了保证 ElAura 平台可以向和欣应用程序提供与和欣系统无差别的内核对象服务,定义了客户程序与内核对象的通信层次<sup>[5]</sup>。共有四个层次:主体层、接口层、列集层、传输层,各层的具体描述如表 1 所述。每个层次提供服务帮助上个层次之间通信,且每个层次对于上个层次来说都是透明可替换的。比如第一层传输层,远程对象服务可以采用消息队列的方式传递数据,而内核对象服务可以采用字符设备文件的方式传递数据,但这两种方式对于第二层列集层的实现来说没有任何影响。

(下转第 2314 页)

数字化接收系统的射频处理前端接收射频信号,其主要功能是把前端射频信号转换为适合于 A/D 采样的带宽中频信号。基频软件接收部分主要完成数字信号的接收处理,而相应的解调分析,频谱测量等后续功能的开发都必须基于上述数字接收机各功能的实现。

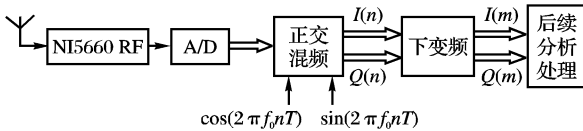


图3 NI5660 接收机结构框架图

我们事先采集一段 NI5660 RF 的接收信号,然后利用该虚拟接收机在本机对系统各个功能,解调算法进行模拟仿真,将其结论用于实际系统的开发中。

运行程序,选用汉宁窗函数滤波,解调算法为 AM,数据采样率  $f_s = 87.5 \text{ MHz}$ ,采样数  $n = 10240$ 。对一段 AM 调制<sup>[5]</sup>,载波频率为 20 MHz 的音频信号进行处理,CIC 抽取因子  $D = 4$ ,输出数据速率为 21.875 MHz。原始音频信号如图 4 所示。

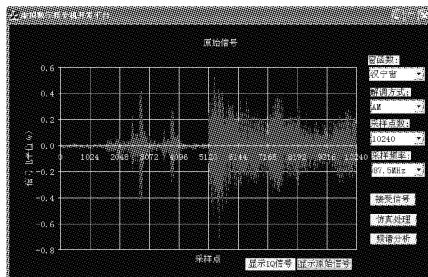


图4 显示原始音频信号截图

对接收信号进行分析处理后的界面截图如图 5 所示,通过观察我们可以发现,经过虚拟接收机处理过后的信号,信号的数据速率明显降低,由 10240 减小到 2560,同时又保持了原有音频信号的基本特征,只是在信号幅值上略有衰减。

除此之外,还对解调信号进行了频谱分析,可进一步分析解调信号的频谱特征。图 6 为信号处理后的频谱分析截图。

#### 4 结语

虚拟数字化接收机系统具有较好的通用性和扩展性,该系统通过纯软件实现,突破了硬件的限制能够更好的研究数字化接收系统的特性,各功能均能很好的得以实现。可以将其应用于实际数字接收系统的开发中去,对现有数字接收系统性能的扩展和改善提供可靠的理论指导。

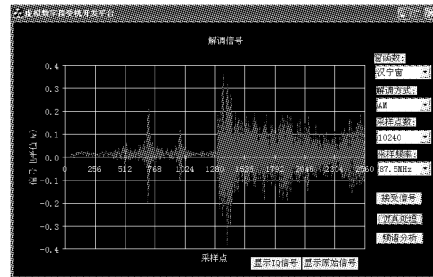


图5 信号仿真处理截图

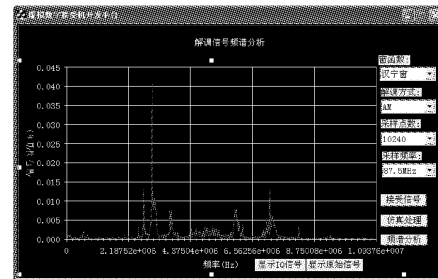


图6 频谱分析截图

#### 参考文献:

- [1] 杨小牛,楼才义,徐建良. 软件无线电原理与应用[M]. 北京:电子工业出版社,2001.
- [2] PUCHER L. Paving paths to software radio design [J]. Communication Systems design, 2001, 7(6): 18 - 27.
- [3] 刘星成,罗恒宏. 软件无线电中的关键技术及平台构建初探[J]. 无线电通信技术,2000,26(1):55 - 59.
- [4] MORELOS-ZARAGOZA R, HARUYAMA S. A software radio receiver with direct conversion and its digital processing [J]. IEICE Transactions on communications, 2002, E85-B(12): 2741 - 2749.
- [5] 刘敏,魏玲. Matlab 通信仿真与应用[M]. 北京:国防工业出版社,2001.

(上接第 2311 页)

表 1 ElAura 平台通讯层

层次	用户空间	内核空间
第四层	客户程序主体, 请求调用内核对象服务	内核对象主体,实现内核对象服务
第三层	以 API 函数和 IKernel 接口向客户程序提供内核对象服务	所有的内核对象都被一个总的 IKernel 接口封装
第二层	标准代理对象, 将客户请求调用信息打包	内核对象的存根、代理、调用对应的内核对象服务
第一层	将请求从用户空间传递到内核空间	将结果从内核空间传递到用户空间

#### 4 结语

对于网络操作系统或网络软件运行平台来说,跨越其他宿主操作系统运行是必要条件。ElAura 构件运行平台提供了一套符合 CAR 构件规范的系统服务构件及支持构件相关编程的 API 函数,实现并支持系统构件及用户构件相互调用的机制,为 CAR 构件提供了运行环境。ElAura 平台使和欣应用

程序能够跨越多种现有的操作系统,一次编译,到处运行,真正实现应用服务的整合和无所不在的服务。其成功实施,使得和欣上层应用整体解决方案与下层的宿主操作系统可以进行适配,便于充分复用积累的经验 and 软件资源。

#### 参考文献:

- [1] CHEN R. The application of middleware technology in embedded OS [C] // Workshop on Embedded System. Conjunction with the ICYCS(6th)2001. Hangzhou: [ s. n. ], 2001.
- [2] BLUNDEN B. Virtual machine design and implementation in C / C++ [M]. Plano: Wordware Publishing, 2002.
- [3] BOX D. Essential COM [M]. 3rd ed. Boston: Addison-Wesley, 1998.
- [4] BRYANT R E, O' HALLARON D. Computer systems a programmer's perspective [M]. Upper Saddle River: Prentice Hall, 2003.
- [5] 苏杭,高■,陈志成,等. Elanix 内核对象服务通信机制的设计与实现[J]. 计算机工程与设计,2006,27(18):3318 - 3329.