

基于XML的嵌入式系统日志解决方案

邱鑫, 林颖, 王保保

(西安电子科技大学 计算机学院, 陕西 西安 710071)

摘要:随着网络技术的发展,XML以其良好的特性在数据交换中扮演着重要角色,文中针对嵌入式系统提出了一种基于XML的日志解决方案,分析了常用的两种XML API,给出了在嵌入式Linux系统下miniXML的移植过程,阐述了使用SAX模型结合miniXML完成XML文档的解析的工作过程。该方案利用XML完成了数据的透明存储,其模块化的设计思想降低了和其它模块的耦合度,具有较高的可移植性和可扩展性。该方案在一款基于华为海思3510的嵌入式网络摄像机的日志模块中运行良好。

关键词:嵌入式Linux; 日志; XML; miniXML; SAX; 移植

中图分类号: TP311.52

文献标识码: A

文章编号: 1673-629X(2009)05-0052-04

An XML - Based Log Solution in Embedded System

QIU Xin, LIN Ying, WANG Bao-bao

(College of Computer, Xidian University, Xi'an 710071, China)

Abstract: With the development of network technology, XML plays a more and more important role in data exchange with its good characteristics. Proposes an XML-based log solution for embedded system, analyses two commonly used XML API, presents the migration of miniXML in embedded system, describes the process of parsing XML documents using miniXML with SAX. The solution realizes transparent storage of data with XML, and its modular design reduces the coupling with other modules with high portability and scalability. The solution has a good performance in the log module of an IP-CAMERA based on Hisilicon 3510 of Huawei Technologies Co., Ltd.

Key words: embedded linux; log; XML; miniXML; SAX; migration

0 引言

嵌入式系统是以应用为中心,以计算机技术为基础,并且软硬件可裁剪,适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统^[1]。作为计算机应用的一个重要领域,嵌入式系统已深入到社会的方方面面,越来越受到人们的关注。而随着数字视频压缩编码的日益成熟和网络技术的发展,以及人们对安防的重视,嵌入式网络摄像机得到了越来越多的应用。一般来讲,PC客户端和嵌入式网络摄像机构成C/S架构,PC客户端通过网络提交请求,对嵌入式设备端进行监视,参数配置等,在此过程中PC客户端的操作需要在嵌入式端记录下来,以形成日志文件今后追查。

传统的嵌入式系统开发中,数据存储主要采用流式文件,如文本格式或某些专用存储格式,其缺点是程

序对数据结构的依赖性强,不利于软件的升级和维护,且软件的可重用性差,采用XML规范进行数据存储和处理可较好地解决上述问题^[2]。文中用一个具体实例来详细说明如何在嵌入式网络摄像机系统的日志开发中使用XML技术。

1 XML及其语言结构

1.1 XML概述

XML(eXtensible Markup Language)即可扩展标记语言,是由W3C于1998年2月发布的一种标准。它是SGML(Standard Generalized Markup Language)的一个简化子集,它结合了SGML和HTML的优点并消除其缺点,以一种开放的、自我描述方式定义了数据结构^[3]。

XML是一种元标记语言,而且支持无限嵌套。由于XML不需要使用任何一种特定的语言并且是以文本形式描述的,因此XML与平台无关,适合于各种平台环境的数据交换。同时,由于基于XML的数据是自我描述的,数据不需要有内部描述就能被交换和处

收稿日期:2008-09-11

作者简介:邱鑫(1983-),男,湖北黄冈人,硕士研究生,研究方向为嵌入式软件与计算机网络;王保保,教授,研究方向为网络数据库与智能检测。

理。作为表示结构化数据的一个工业标准,XML为组织、软件开发者、Web站点和终端使用者提供了许多有利条件^[4]。

1.2 XML语言结构

每个XML文档都由XML声明开始,一般XML文件的第一行便是XML声明,例如:`<?xml version="1.0" encoding="UTF-8"?>`,它指定文档中使用的XML的版本和字符集。XML中最常用的标签由包围在一个小于号(<)和一个大于号(>)之间的文本组成,例如`<tag>`。起始标签(start tag)表示一个特定区域的开始,例如`<start>`;结束标签(end tag)定义了一个区域的结束,除了在小号之后紧跟着一个斜线(/)外,和起始标签基本一样,例如`</end>`。SGML还定义了标签的属性(attribute),它们是定义在小号和大号之间的值,例如`<log type="Alarm">`中的type属性。

2 嵌入式系统与XML解析器的移植

2.1 嵌入式系统

嵌入式系统是针对具体应用的专用系统,一般具有成本敏感性。其硬件和软件都必须进行高效率的设计,量体裁衣、去除冗余,力争在同样的硅片面积上实现更高的性能^[5]。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户应用程序等四个部分组成,用于实现对其他设备的控制、监视或管理等功能。

2.2 嵌入式的XML解析器

嵌入式的XML解析器是XML在嵌入式系统中应用的关键。XML解析器负责对XML进行解析,用户程序使用XML API访问XML文档,并提取其中的信息。目前,有2种主流的XML API被广泛使用:文档对象模型DOM(Document Object Model)和用于XML的简单API(Simple API for XML, SAX)。

基于DOM模型的解析器在分析时,一次性对整个XML文档进行分析,并在内存中形成对应的树结构,用户代码使用DOM提供的一系列接口来操作这个树结构^[6]。而SAX是一种由事件驱动的XML解析器,在解析到开始标记、结束标记、文本等内容的时候会通知或调用用户程序进行处理。

由于采用了不同的处理方式,DOM占用内存大,速度往往较慢,但具备持久性且可以给用户提供良好的访问接口,对用户更为友好;SAX占用内存小,解析速度快,但不具备持久性且需要用户程序负责具体的处理^[7]。

在文中所述的实现中,采用miniXML作为实现嵌

入式的XML解析器。miniXML是使用C语言编写的,其开发初衷就是以一种够小且易用的解析器取代相对庞大的libxml2^[8],难能可贵的是,它提供DOM和SAX两种接口。

miniXML已经被广泛应用于若干开源项目中,包括Gutenprint, ZynAddSubFX等。miniXML使用C语言开发,与操作系统无关,因而能够很好地移植到多种嵌入式平台下。

2.3 miniXML在嵌入式Linux下的移植

首先搭建好交叉编译环境,在文中所述的实现中,嵌入式设备端的Linux内核版本为2.6.14,安装的交叉编译工具的版本为arm-linux3.4.3。

然后到miniXML的主页(<http://www.minixml.org>)下载源码包。目前可下载到的最新的版本为2.5。将下载的源码包解开,生成mxml-2.5目录。

接着进入mxml-2.5目录,运行./configure配置文件进行交叉编译所需Makefile文件的生成,在生成之前需要修改配置参数,具体的配置参数可以通过运行./configure -help进行查看,文中所述的实现中,配置步骤如下:

①添加交叉编译指示: `--host=arm;`

②调整C编译器: `CC=arm-uclibc-linux-gcc;`

③设置C编译指示: `CFLAGS=-I/opt/hisilicon/toolchains/sysroot-uclibc-soft/usr/include;`

④设置链接指示: `LDFLAGS=-B/opt/hisilicon/toolchains/sysroot-uclibc-soft/usr/lib。`

运行./configure,后面跟上按如上步骤设置好的参数,配置完成后,当前目录下会生成一个交叉编译需要的Makefile文件。打开此Makefile文件,继而修改AR=arm-uclibc-linux-ar, RANLIB=arm-uclibc-linux-ranlib,保存完后执行make命令就可以在当前目录生成静态库文件libmxml.a,依据这个库文件,结合源码包里面提供的mxml.h头文件,就可以进行XML解析相关的程序设计。

3 日志方案的具体实现

3.1 XML的构建

文中所述的平台是一个基于嵌入式Linux的网络摄像机,嵌入式设备端采集音视频信息,PC客户端通过网络连接登录到设备端进而进行画面监控、设备端参数配置、系统升级、重启等操作,用户的操作都需要记录到设备端。系统需要保存的日志信息见表1。

实现过程中,根据上表中的数据,定义一个XML文件来进行系统日志的保存,对于XML文件的构建形式,考虑了如表2所示的2种方式。

表 1 日志信息列表

日志项目	是否必须	备注
日志 ID	是	每条日志的 ID 唯一
时间	是	客户端用户某项操作的时间
用户	是	客户端登录用户名
用户组	是	客户端登录用户所属的组
事件类型	是	相当于客户端做的什么操作
日志内容	否	客户端具体操作内容

表 2 XML 构建形式

方式 1	方式 2
<pre><? xml version = "1.0" encoding = "UTF-8"? > <all> <log id="1" tm="2008-8-8" nm="admin" gp="admin" tp="Alarm" ct="" /> </all></pre>	<pre><? xml version = "1.0" encoding = "UTF-8"? > <all> <log> <id>1</id> <tm>2008-8-8</tm> <nm>admin</nm> <gp>admin</gp> <tp>Alarm</tp> <ct></ct> </log> </all></pre>

表中,tm 代表时间,nm 代表用户名,gp 代表用户组,tp 代表事件类型,ct 代表日志内容。观察表 2,相同的一条日志内容,方式 1 对比方式 2 的文件开销更小,更利于日志文件的存储,减小系统空间的占用。

3.2 具体的解析及封装

本嵌入式系统仅 16M RAM,在设备端服务器进程申请内存缓冲区启动音视频捕获并从系统分配大小为 4M 的内存池后,内存空间所剩不多,故考虑使用 DOM 模型进行 XML 文件的解析不是很实际,且笔者尝试在嵌入式设备端启动服务器进程并开启音视频捕获任务后,采用 DOM 模型对一个有上万条日志的 XML 文件进行解析,解析速度很慢,而且会出现解析过程中出现 kill 掉服务器进程的现象,直接对系统的核心功能造成影响,所以从减小系统资源占用和从不影响设备端服务器进程出发,基于 miniXML 的 SAX 模型来进行对 XML 文件的解析。

解析前先调用 mxm1SAXLoadFile 加载文件,调用过程中设置其中的 sax_cb 参数为用户自定义的 SAX 回调函数。回调函数的原型为: void sax_cb(mxm1_node_t * node, mxm1_sax_event_t event, void * data), 其中的 node 是目前处理到的 XML 文件中的一个节点, event 的参数是具体的事件类型,它是回调函数需要处理的关键,根据 event 的类型分析节点,并将需要

的节点保存到节点树,解析结果在最终的节点树中。具体的解析流程如图 1 所示, event 的具体类型及其说明如表 3 所示。

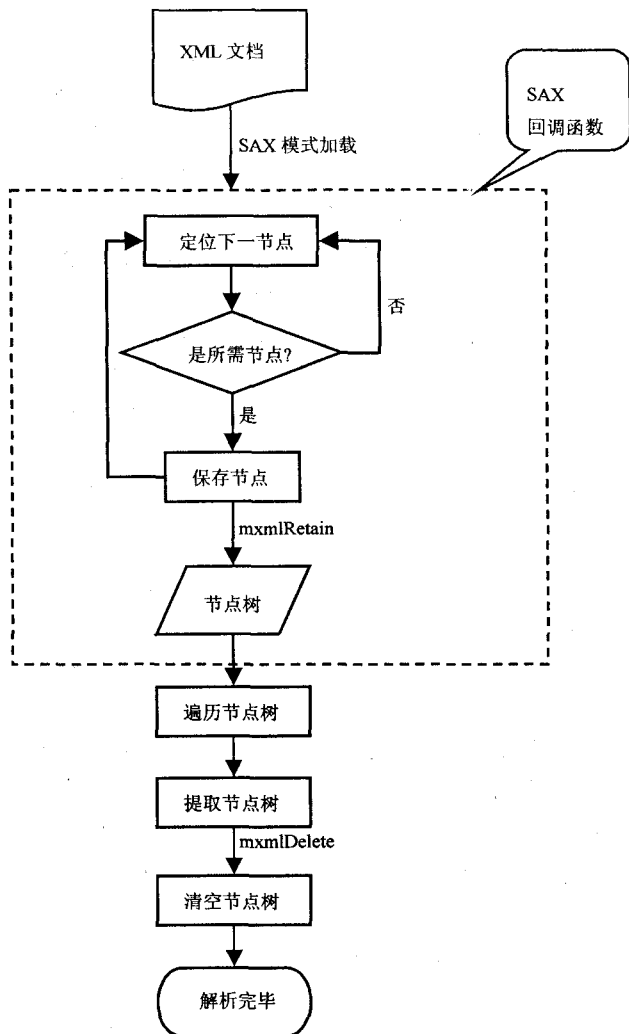


图 1 XML 文件解析流程

表 3 event 的具体类型及其说明

event 类型	event 说明
MXML-SAX-CDATA	读到 CDATA
MXML-SAX-COMMENT	读到注释
MXML-SAX-DATA	读到数据(文本等)
MXML-SAX-DIRECTIVE	读到处理指示(例如 XML 文件声明等)
MXML-SAX-ELEMENT-OPEN	读到节点开始标记
MXML-SAX-ELEMENT-CLOSE	读到节点结束标记

miniXML 提供了解析一个 XML 文档所需的若干函数。主要包括:

- ①文件加载: mxm1LoadFile, mxm1SAXLoadFile, 其中前者使用 DOM 模型,后者使用 SAX 模型。
- ②读取和设置节点属性: mxm1ElementGetAttr, mxm1ElementSetAttr。

③节点的查找和遍历: mxmFindElement, mxmWalkPrev, mxmWalkNext。

④产生节点和文本段 mxmNewElement, mxmNewText。

⑤文件保存: mxmSaveFile, mxmSaveFd。

⑥保存和删除节点: mxmRetain, mxmDelete。

以上列出的只是笔者在实现过程用的比较多的几个函数, miniXML 提供了一系列丰富的接口供用户调用, 更多的函数细节可参考 miniXML 官方说明文档^[8]。

为降低设备端服务器应用程序开发过程中的系统复杂度和模块之间的耦合度, 设计一个专门负责日志处理的 CLog 类, 将对日志文件的解析和写入在类中进行封装。

由于系统对日志的处理分布于不同模块的不同地方, 故这里考虑对 CLog 类仅创建一个全局对象, 由它专门来负责日志添加、查询等操作。这里使用 Singleton(单例)模式, Singleton 模式解决问题十分常见, 在基于对象的设计中可以通过创建一个全局对象来实现, 在面向对象和面向过程结合的设计范式(如 C++)中, 也可以通过一个全局变量实现这一点^[9]。Singleton 模式典型的结构图如图 2 所示。

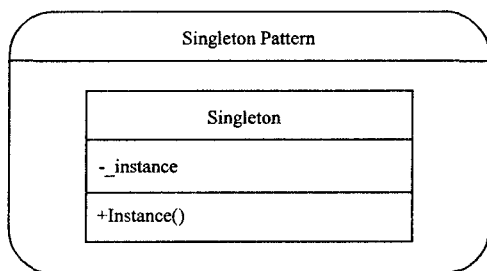


图 2 Singleton Pattern 结构图

CLog 类对提供的 AddLog() 方法负责一条日志的添加, 在需要添加日志的地方调用 CLog::Instance() -> AddLog(), SearchLog() 负责解析 XML 日志文件, 它在按如图 1 所示的解析节点的过程中, 根据客户端传递过来的日志查询条件信息, 将符合查询要求的节

点添加到节点树, 最后遍历最终的节点树将结果文件传递给客户端, 客户端据此 XML 文件进行解析, 查询日志时调用 CLog::Instance() -> SearchLog(), 即得到查询结果。CLog 类封装了写入和解析 XML 的具体操作, 采用了模块化的结构, 降低了和其它模块的耦合度, 实现了数据的透明存储, 具有较好的移植性。

4 结束语

文中结合 XML 等技术, 提出了一种基于嵌入式 Linux 系统的日志解决方案, 并利用 miniXML 实现具体功能。为降低系统复杂度和与其它模块的耦合度, 该方案采取了模块化的思想。基于 XML 的机制实现了对日志数据的透明存储, 提高了代码的可移植性和可扩展性。

参考文献:

- [1] 任伟, 张毅, 聂捷楠. 基于嵌入式系统开发平台的设计与实现[J]. 微计算机信息, 2006, 22(10): 178-179.
- [2] 王婷, 陈和平. XML 技术在嵌入式系统图形界面开发中的应用[J]. 武汉科技大学学报: 自然科学版, 2005, 28(1): 75-77.
- [3] World Wide Web Consortium. Extensible Markup Language [EB/OL]. 2008-07-06. <http://www.w3.org/XML/>.
- [4] 吕锋, 余丽. 基于 XML 的 Web 数据抽取研究[J]. 计算机技术与发展, 2007, 17(6): 53-54.
- [5] 程广河, 郝风琦, 张让勇, 等. 嵌入式环境中的软件构件化研究[J]. 计算机技术与发展, 2007, 17(9): 139-140.
- [6] Idris N. Should I use SAX or DOM? [EB/OL]. 1999-05-23. <http://developerlife.com/saxvsdom/file.pdf>.
- [7] 袁菲, 陆阳. XML 在嵌入式环境下的应用[J]. 合肥工业大学学报, 2007, 30(4): 445-448.
- [8] Sweet M. Mini-XML Programmers Manual, Version 2.5 [EB/OL]. 2008-01-28. <http://www.minixml.org/mxml.html>.
- [9] Eckel K. Design Pattern Explanation with C++ Implementation [EB/OL]. 2005-04. <http://www.mscenter.edu.cn/blog/k-eckel>.

(上接第 51 页)

- [5] 史册, 徐胜荣, 荆仁杰, 等. 实时图像处理中一种快速的直线检测算法[J]. 浙江大学学报: 工学版, 1999, 33(5): 482-486.
- [6] Canny J. A Computational Approach to Edge Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 8(6): 679-698.
- [7] Freeman H. Boundary encoding and processing[M]. New

York: Academic Press, 1970: 241-266.

- [8] Yuan Jianxing, Suen C Y. An optimal O(n) algorithm for identifying line segments from a sequence of chain codes[J]. Pattern Recognition, 1995, 28(5): 635-646.
- [9] Andrew H, Lai S, Nelson H, et al. Lane detection by orientation and length discrimination[J]. IEEE Transactions On Systems, Man and Cybernetics, 2000, 30(4): 539-548.