

Darwin Streaming server 的研究与应用

黄拔峰¹, 钟 明¹, 杨传钧², 张家钰¹

(1 东华大学计算机科学与工程系, 上海 200051; 2. 上海竞天科技股份有限公司, 上海 200120)

摘 要: 分析了Darwin streaming server(DSS)的架构和核心流程; 讨论了Darwin streaming server的二次开发接口: module; 最后给出了一个用户认证的二次开发模块。

关键词: 流服务器; 流媒体; MPEG4; QuickTime; 达尔文

Research and Application of the Darwin Streaming Server

HUANG Bafeng¹, ZHONG Ming¹, YANG Chuanjun², ZHANG Jiayu¹

(1. Department of Computer Science and Engineering, Donghua University, Shanghai 200051;

2. Shanghai Gentek Corporation Ltd., Shanghai 200120)

【Abstract】This article analyzes the architecture and the core flow of the Darwin streaming server(DSS), then discusses the so-called module, namely the interface of the redevelopment of the Darwin streaming server, at last it shows an example about the development of a user authentication module.

【Key words】Streaming server; Streaming media; MPEG4; QuickTime; Darwin

随着宽带网络基础设施的不断完善,流媒体应用的广度与深度都在发生根本的变化。目前的流媒体领域,主要有3种平台: RealNetworks公司的RealSystem, 微软公司的Windows Media, 苹果公司的QuickTime。3大平台都有相应的流服务器作为其应用的核心, 其中苹果公司的流服务器是QuickTime Streaming Server(简称QTSS)。QTSS同时支持FreeBSD、Linux、Solaris、Windows NT和Windows 2000等多个操作系统, 是当前所有同类产品中支持平台最多的一个。目前QuickTime文件格式已被国际标准组织(ISO)选为MPEG-4的基本文件格式。另外苹果公司作为Internet流媒体联盟(ISMA)的核心成员不但其流媒体的技术与解决方案符合标准, 而且QTSS是开放源代码的, 其项目名称是Darwin streaming server(下文简称DSS)。开发人员可以在苹果公司的网站上获取相关的源码和文档。

1 DSS的整体架构

从应用角度, DSS可以抽象为两大系统: 一个是文件处理系统, 另一个是服务器核心。如图1。



图1 DSS的架构

1.1 文件处理系统

文件处理系统负责将线索化(Hinted)过的QuickTime文件或ISO MPEG4文件通过RTSP和RTP协议流化出去。所有分析这些文件的代码都被提取出来并且封装在QTFile库中。

目前QTFile可以处理QuickTime文件和ISO MPEG4文件。开发人员如果希望让DSS支持其他媒体格式, 则需要针对相关格式开发文件流化工具。

1.2 服务器核心

处理网络 and 协议。主要有3个子系统: RTSP子系统, RTP子系统以及公共服务子系统。

(1) RTSP子系统

负责解析和处理RTSP请求。每一个RTSP/TCP连接都对应一个RTSP的session。主要的类有RTSPSession, RTSPRequest, RTSPResponseStream和RtspRequestStream。

(2) RTP子系统

负责媒体数据包的发送, 根据RTCP的反馈进行服务质量控制。主要的类有RTPSession, RTPStream和RTCPTask。

(3) 公共服务子系统

负责服务器的启动/关闭、初始化参数设置以及为module机制、跨平台的多线程和事件机制等提供支持。该子系统的类的前缀都是“QTSS”。

DSS提供了一种称为module的二次开发接口。使用这个开发接口, 我们可以自由扩展服务器的功能。DSS中的许多核心功能也是以module的方式预先实现并且编译的, 在下文的分析中可以看到DSS的许多任务都是通过调用相应的module来完成的。

关于module在第3节中详细描述, 这里先介绍一下Role的概念。一个Role就是一项任务, 一个module可以注册若干个Role, 表示这个module可以处理这些注册的任务。如一个module注册了QTSS_RTSPPreProcessor_Role, 则可以调用这个module来处理RTSP请求。

2 核心流程描述

选取这样一个流程: 从client发出“play”请求, server接收、分析、处理该请求, 最后发送数据包。这个过程基本反应了server端的核心流程。下面以这一流程处理为例作简要介绍。

2.1 RTSP请求分析

RTSP可以理解为一个远程的VCR控制协议, RTSPSession类是用来处理客户请求的。当client端发出Play

作者简介: 黄拔峰(1978—), 男, 硕士生, 主要研究方向: 流媒体; 钟 明, 硕士生; 杨传钧, 硕士; 张家钰, 教授、硕士
收稿日期: 2003-07-01

请求时，server端的RTSPListenerSocket监听到这个请求，创建一个RTSPSession，这个RTSPSession被加入到任务队列中，当时间片到达时，TaskThread线程就会调用RTSPSession对象的Run函数，在Run函数中，维护一个RTSPSession状态机，对客户的RTSP请求做出不同的处理。请求分析是在“过滤请求状态”(kFilteringRequest)下完成的。在该状态下，调用SetupRequest解析该请求并找到相应的RTPSession。

2.2 RTSP请求处理

请求分析完成后，RTSPSession进入请求处理状态(kProcessingRequest)，DSS会调用注册了“请求处理任务”(QTSS_RTSPRequest_Role)的module，而QTSSFileModule就是这样一个module。QTSSFileModule中定义了一个分发函数QTSSFileModuleDispatch，它根据传入的任务类别和任务参数调用相应的函数。本例中传入的任务是QTSS_RTSPRequest_Role，相应的处理函数是ProcessRTSPRequest，该函数根据传入的RTSP method（详见RFC2326）调用相应的处理函数，本例传入的method是play，所以调用函数DoPlay。DoPlay实际调用了RTPSession的Play函数。RTPSession的Play函数为播放做了很多准备工作，如缓冲区的设置等，在该函数的最后调用Signal方法，该方法将RTPSession加入到任务队列中，当时间片到达时，RTPSession的Run方法被激活。

2.3 发送RTP数据包

RTPSession的Run函数调用QTSSFileModule，该module的sendpackets函数解析文件，读取数据，生成RTP包，写入RTPStream，最后RTPStream调用socket，向客户端发送数据包。

2.4 流程的序列图

图2展示了这个流程，限于篇幅，对于不是很关键的内容，如与socket相关的类都省略了。

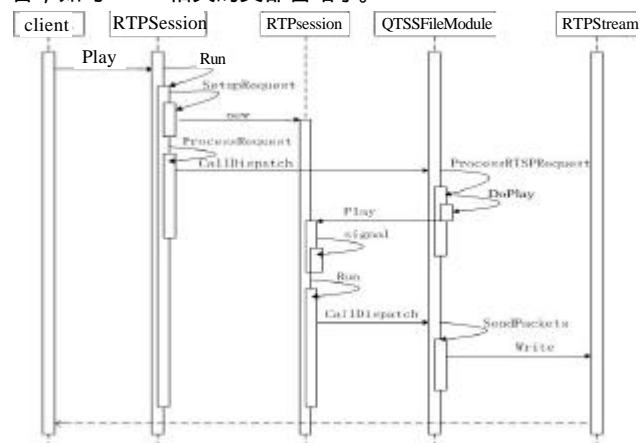


图2 server的处理流程

3 DSS二次开发接口：Module

DSS的二次开发接口叫做module。它的规范非常简单，读者只要熟悉C语言，就可以非常容易地开发一个module。当然这必须以真正理解DSS的架构和流程为前提，否则不可能开发出实用的module。

下面通过分析DSS的一个内嵌module：QTSSFileModule的源代码来说明Module的开发方式，QTSSFileModule在QTSSFileModule.cpp文件中实现，每个module必须实现两个函数：主函数和分发函数。

服务器调用module的主函数用于启动和初始化模块中的QTSS函数。QTSSFileModule主函数的实现如下：

```
QTSS_Error QTSSFileModule_Main(void* inPrivateArgs)
{
    return _stublibrary_main(inPrivateArgs,
    QTSSFileModuleDispatch);
}
其中QTSSFileModuleDispatch就是module必须实现的分发函数名。服务器调用分发函数实现某一类任务。分发函数根据传入任务的名字和一个任务相关的参数块调用相应的处理函数。QTSSFileModule分发函数的完整代码如下：
QTSS_Error QTSSFileModuleDispatch(QTSS_Role inRole, QTSS_RoleParamPtr inParamBlock)
{ switch (inRole) //任务名称
{ case QTSS_Register_Role:
    return Register(&inParamBlock->regParams);
    //相应的处理函数
case QTSS_Initialize_Role:
    return Initialize(&inParamBlock->initParams);
case QTSS_RereadPrefs_Role:
    return RereadPrefs();
case QTSS_RTSPRequest_Role:
    return ProcessRTSPRequest(&inParamBlock->
    rtspRequestParams);
case QTSS_RTPSendPackets_Role:
    return SendPackets(&inParamBlock->
    rtpSendPacketsParams);
case QTSS_ClientSessionClosing_Role:
    return DestroySession(&inParamBlock->clientSessionClosing
    Params);
}
return QTSS_NoErr;
}
```

DSS提供了两种方式把我们自己开发的module添加到服务器中：一种称为静态模块（Static Module），该方式将我们开发的module代码直接编译到内核中去；另一种称为动态模块（Dynamic Module），该方式将我们开发的Module单独编译称为一个动态库，然后修改配置，使服务器在启动时将其加载。

当服务器启动时，它首先装载没有被编译进内核的动态模块，然后才装载被编译进内核的静态模块；由于现有的大部分系统功能都是以静态模块的方式存在的，如果你希望用自己的模块替换某个系统功能，最好是编写一个动态模块，因为它们将早于静态模块被装载。无论是静态模块还是动态模块，它们的代码都是相同的，唯一的不同就是它们的编译方式。

首先为了将静态模块编译到服务器中，必须修改QTSServer.cpp文件中QTSServer::LoadCompiledInModules，并向其中加入以下代码：

```
QTSSModule* myModule=new QTSSModule(*_XYZ_*);
(void)myModule->Initialize(&sCallbacks,&_XYZMAIN
);
```

(void)AddModule(MyModule);

其中，XYZ是静态模块的名字，而XYZMAIN则是其主函数入口。

动态模块的编译方法如下：首先单独编译动态模块为一个动态共享库；将该共享库与QTSS API stub library 链接到

（下转第143页）

参考文献

- 1 李凤保, 刘金, 古天祥. 网络化传感器技术研究[J]. 传感器技术, 2002, (7): 62-64
- 2 李凤保, 古天祥. 网络化仪器技术研究[J]. 自动化仪表, 2003, (5): 62-64
- 3 International Electrotechnical Commission. IEC 61158 parts 3 to 6: Digital Data Communications for Measurement and Control

(上接第135页)

一起;最后将结果文件放置到特定目录下,如在Windows上,放到DSS安装目录的QTSSmodule文件夹下。此后,服务器在启动时就将自动调用该动态模块。

4 结束语

作为一个具有工业强度的产品,同时又是一个不断发展的开放源码项目,DSS为学习应用流媒体提供了一个很好的平台。在掌握其基本设计框架和基本流程的基础上,读者可以根据需要深入源码,去挖掘流媒体技术的精华,针对具体

(上接第140页)

和硬件的数据交换。VxWorks应用过程中首先要完成硬件相关代码,硬件相关代码中的板级支持包BSP和硬件驱动代码是底层开发的核心,硬件相关代码完成后,VxWorks操作系统和硬件之间的接口就已经完成,在此基础上即可进行应用程序的开发。

2.1 底层开发

VxWorks操作系统的底层开发包括板级支持包BSP的设计和驱动的设计,这是一个相对复杂的过程。VxWorks的板级支持包BSP(文献[4]p310)提供VxWorks同硬件环境的基本接口界面,负责在上电时的硬件初始化,支持VxWorks对硬件驱动的访问,将VxWorks中硬件相关和硬件无关的软件集成为一体,其功能上相当于PC机的BIOS。BSP的文件构成包含源文件(多为C语言源程序少量汇编程序),头文件(包含文件),make文件(控制映像文件images的生成),导出文件和二进制的驱动模块。VxWorks的BSP应和WRS(WindRiver System)的产品兼容,因而最好有一个可参考的BSP模板,在此基础上修改相关的文件。并通过修改Config.h和Makefile使相应的修改和原有的代码连接到一起。关于BSP的详细流程和设计方法^[1]。

VxWorks的缺省配置提供大量通用接口的驱动程序,但是这些通用接口的驱动程序的参数可能并不适合你的硬件环境,你必须使用ar d命令(对于Power CPU为ar PPC d)将这些模块从库文件中删除,然后将其源程序复制到你的BSP目录中进行改动,并在Makefile中使用MACH_EXTRA命令将其目标模块加入到最终的VxWorks映像文件中。如果你的硬件中含有专用接口,你需要自行编制这些专用接口的驱动程序,并在Makefile中使用MACH_EXTRA命令将其目标模块加入到最终的VxWorks映像文件中。

本系统的底层软件包括CPU相关的初始化,如CPU内核、SDRAM接口、ROM接口、FPU、PCI主桥等;PCI总线设置,包括PCI9030桥片和DDC62864的PCI配置寄存器设置等;通用硬件初始化,如中断控制器、定时器、串口、网口等的初始化。由于本系统支持双冗余的CPU板,只有处于主设备的CPU才能在上电之初对PCI总线进行设置。

Fieldbus for Use in Industrial Control Systems[S], 2000-01

- 4 International Electrotechnical Commission. IEC 61158-6: Digital Data Communications for Measurement and Control---Fieldbus for Use in Industrial Control Systems---part 6: Application Layer Service Definition[S]. type 4, 2000-01
- 5 International Electrotechnical Commission. IEC 61158-6: Digital Data Communications for Measurement and Control---Fieldbus for Use in Industrial Control Systems---part 6: Application Layer Protocol Specification[S]. type 4, 2000-01

需求进行二次开发。DSS非常庞大,本文只是起到抛砖引玉的作用,更多的资料可以到苹果公司的网站上去寻找。另外,在学习和开发过程中,DSS的邮件组也可以提供更多的实际帮助。

参考文献

- 1 共创软件. Apple公司Darwin流式服务器源代码分析 [EB/OL]. <http://cosoft.org.cn/html/publish/>
- 2 Apple文档. QuickTime Streaming Server Modules [EB/OL]. <http://developer.apple.com/darwin/projects/streaming/>

2.2 应用程序

基于VxWorks操作系统的应用程序开发是在VxWorks集成开发环境tornado中完成的,有关tornado的使用请参考文献[3]。

本系统的应用软件主要是1553B接口的软件,1553B接口采用DDC公司的62864芯片,可工作于BC(Bus Controller)、RT(Remote Terminer)、MT(monitor)3种方式。1553B接口软件采用任务+中断服务程序的方法实现1553B的总线通信。中断服务程序通过调用相应的接收数据处理子程序或向通信管理任务发出信号的方法完成接收数据的处理,在接收数据发送结束后处理相关的协议。

另外为了适应星载软件在轨软件维护的要求,同时便于应用软件的开发,软件系统采用了Vxworks的TFFS文件系统,并利用VxWork提供的模块动态管理库函数,实现软件模块的在线改写,动态改变软件的运行,实现应用软件的动态维护。

为了支持对系统软件的动态维护,硬件上设置了两冗余的BootROM。在更换系统代码时,先更换备份的BootROM,完成后在备份的BootROM上重新引导系统,重新引导完成后,再更换主份的BootROM。

3 小结

本文介绍了一种以MPC8240 CPU和VxWorks实时多任务操作系统为核心的嵌入式计算机系统的设计与实现过程,重点介绍了硬件系统的结构,可靠性措施和硬件设计中应注意的问题,同时介绍了软件的底层开发和应用程序开发的过程。

参考文献

- 1 Tornado BSP training workshop[M]. Imagination Corporation, 2000.
- 2 Tornado Driver training workshop[M]. Imagination Corporation, 2000
- 3 Tornado training workshop[M]. Imagination Corporation, 2000
- 4 VxWorks 5.4 programmers guide[M]. WindRiver Corporation (Edition), 1999-04-09