Thinking beyond Source Code

Remember Steve Jobs

- Home
-
- Emacs Configuration
- Moizlla Plugin
- About Me
- Shift Scheduler Support

Home androidAndroid Audio System(2): ALSA Layer

Android Audio System(2): ALSA Layer

三月 15th, 2010 0 Comments/4903 hits

*** Hardware Layer: Alsa**

Author : Zhang Jiejing (http://www.thinksrc.com)
Data: 2010-3-15

ALSA hardware layer is becoming a standard part of Android master tree.
the Android ALSA layer library Code base:
http://android.git.kernel.org/?p=platform/hardware/alsa_sound.git;a=summary
The major development job is by WindRiver.
This paper is a continue part of the past post Audio audio system(1), if you haven't read it , I strongly suggest you read it first.

**** Roles**

***** AudioSystem - Audio Abstract Layer**

code: frameworks/base/media/libmedia/AudioSystem.*

***** AudioPlicyManager - Platform specific policy manager**
Alsa: AudioPolicyManagerALSA.* : This Manager can be implement in OSS or ALSA.
***** AudioStream[In|Out]Alsa**
This class is controled by AudioPolicyManager, charge the read and audio mix job. using alsa pcm interface.
***** Alsa device handler: reference code: alsa_default.cpp**
This part is compiled as a separated shared library, alsa_default.cpp just acts as a fall back.(This is just like other libhardware modules, the libhardware frist search the [ModuleName].[ro.hardware prop].so such as alsa.dream.so and ro.product.board, ro.board.platform, ro.arch is continue search, if can't find any of these libraries, it's will fall back to search [ModuleNmae].default.so, such as alsa.default.so)
**** Communation betweens Roles:**
AudioSystem has a AudioPolicyServer Client, it will PRC to AudioPlicyManager class, in this case, the class is AudioPolicyManagerALSA.
AudioStream[In|Out] will use hardware lib via dlopen() the ALSA handler library. And the call the open/close directly.

**** Control follow:**

***** Media Server side.**

When recording, the AudioSystem will call AudioSystem::getInput first, this call will deliver to AudioPolicyManagerALSA::getInput(); AudioPolicyManagerALSA::getInput() will check channel, input format, sampleRate, and call the mpClientInterface openInput's interface(AudioPolicyManagerALSA.cpp:1014), this call will back into audio_flinger 's openInput(AudioPolicyServer.cpp:504),

***** Back to AudioFlinger**

In AudioFlinger::openInput,This function will wil call AudioHardware's openInputStream(), this will drop into AudioHardwareALSA class(audioHardwareALSA.cpp:229), this fucntion will call the hardware list and call the open function of this device, this will into the hardware shared library(such as alsa_default.cpp) 's open function.

***** Into hardware handler library**

In the open() it will call snd_pcm_open, this is a ALSA user library function, this function will do some check device, and apply the config (such as hooks in asound.conf) belows to this device. Also it's will do some params setting such as set Hardware params, and software params.After open() returns, the the AudioHardwareALSA will new a AUdioStreamInALSA class, by the handler open() returns, and acoustics.

***** Leave hardware handler library**

In AudioStreamInALSA, it will init the ASLAStreamOps class by the alsa hardware hander(for example, alsa_default).Also will create a acousitic device in same time.After that, AudioHardwareALSA will call a stream's set(), this will be in the ALSAStreamOPS::set() function.This function will see the channel config in hardware handler, and format and sample rate, and perpare for the parameter AudioHardwareALSA need.These paramenter will finnally return to audioFlinger, audioFlinger will check whether these valuse are same to what he want, if the value is not same and not by error, the audioFlinger will call openInputStream again by these paramenters.

If the the input is open success, the a RecordThread will be created, and put it into the record thread pool.And then, the record precess will same to other non-ALSA sound hardware.This is how the ALSA connect to AudioSystem and how ALSA connect to ALSA hardware handler.

android

Related Post

- Linux Kernel and Android Suspend/Resume
- Android: 移植VPN相关的kernel 内容
- Linux Kernel and Android 休眠与唤醒(中文版)
- Android Audio System (1)
- android的域名解析

Comments (0) Leave a comment

Name (required)

Mail (will not be published) (required)

Website

=9+8 (required)

Submit Comment (Ctrl+Enter)

☑ Notify me if there is a reply

迁移到GAE上面 在ubuntu中添加debian的源并安装xtables-addons
想订阅这个博客

Recent Posts

- 关于电信光纤不能上pptp的解决办法
- 配置PPTP VPN服务器的一些注意事项
- Debug Android and Linux suspend and resume （中文）
- Steve Jobs, You Are Living Forever In My Heart
- Linux技巧： mount 同一个目录到两个地方
- Use debugfs tracing in linux kernel.
- 用go写的一个dd程序
- How to Use Office Communicator under Ubuntu

Recent Comments

- Retin-A on Emacs Configuration
  comment1, Propecia, »
- Skiller on Android Audio System (1)
  "Application -& »
- kzj on Moizlla Plugin
  ok, I'm asking goog »
- anonymous on Moizlla Plugin
  Your Google SSL is n »

Ray on Android Audio System (1)
"if can't get a »

-

Meta

- Login

Tags

android bluetooth linux pm uart GAE GFW Linux android apple emacs git go internet linux linux 优化 ubuntu 性格 电子商务

Blog roll

- 徐明的博客
- 4G
- 8gdns
- 叶歆昊的个人博客
- e账本 - 简单的记账软件

Top