

MIDLinux 新一代播放器的设计与实现

潘影波¹,王 健¹,Rusty Lynch²

(1. 东南大学 计算机科学与工程学院, 江苏 南京 211189; 2. Intel 亚太研发有限公司, 上海 200241)

摘 要:随着 Intel 的 MID 移动平台的推出, MIDLinux 正在变得越来越成熟。基于 MIDLinux, 提出了体系结构更加开放的播放器架构。该架构以 Dbus 为通讯桥梁, 以现有的开源播放器 helix-player 和 gstreamer 为起点, 将两者合二为一, 目的是在 Intel 未来基于 Linux 的移动 MID 平台上, 为用户提供功能更加强大的媒体播放能力。

关键词: MIDLinux; helix-dbus-server; gstreamer; helix-player 播放器

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2009)02-0202-03

Design and Realization Intel MIDLinux New Generation Player

PAN Ying-bo¹, WANG Jian¹, Rusty Lynch²

(1. School of Computer Science and Engineering, Southeast University, Nanjing 211189, China;

2. Intel Asia-Pacific R & D Ltd, Shanghai 200241, China)

Abstract: Along with Intel launches its new MID project, MIDLinux is becoming more and more mature. Based on MIDLinux, presents the open framework architecture of media player, the framework will combine two popular players (helix-player and gstreamer) in one, and all its components will communicate on Dbus. Designing this player's aim is to provide users with more powerful media playback capability on Intel's future MID mobile platform.

Key words: MIDLinux; helix-dbus-server; gstreamer; helix-player player

0 引 言

MID 全称 Mobile Internet Device, Intel 08 平台软硬件的全称, 是 Intel 专门为手持设备而开发的, 目的是增强移动设备的网络访问能力。笔者亲自参与并设计实现了 Intel MIDLinux 体系结构下更加开放的播放器 moblin-media, 目的是支持 OSV (operation system vender), 如 redflag 等开发更加符合自己特色的播放器, 同时也为 Linux 开源社区提供功能更加强大齐全的播放器。MID 项目一发起, 就得到了众多 OSV 的支持, 包括 redflag, ubuntu 等 Linux 操作系统提供商的强有力支持, 因而发展迅速, 也是 Intel 对开源软件的又一大贡献。

1 当前 Linux 开源社区播放器的现状

1.1 现 状

现在 Linux 社区开源优秀的播放器很多^[1], 如

mplayer, helix-player, gstreamer。mplayer 是传统的播放器, 历史悠久, helix-player 由于得到了 realnetwork 公司的大力支持, 也赢得了很多用户, gstreamer, 其前端又称 totem, 是一个完整意义上的开源播放器, 由于其体系结构的开放先进性, 也得到很多 Linux 爱好者的青睐, 但是由于没有得到有力的商业支持, 其支持的 mime type 类型也相当有限, 例如 gstreamer 就不支持 rmvb 格式的媒体文件^[2]。再看得得到 realnetwork 大力支持 helix-player, 虽然支持多种格式的 mime type 媒体文件^[3], 但 OSV 若想在其上开发适合自己特色的播放器, 难度很大, 几乎是不可能, 而如果 OSV 是基于 gstreamer 开发自己的播放器, 难度几乎是零, 因为 gstreamer 本来就是基于 Dbus 的开放体系结构的播放器^[4], 其设计的初衷就是给他的用户留有足够的空间和接口供开发者按照自己的意愿做二次开发。让 OSV 的元素融入到播放器中, 这也是大多数 OSV 想看到的结果。最重要的一点是, 各个播放器的播放能力都是有限的, helix-player 也不是万能的, 它就不能播放 swf 格式的文件^[3]。既然每个播放器都存在缺陷, 又有各具所长, 那能不能使其结合, 达到优势互补呢?

收稿日期: 2008-06-22

作者简介: 潘影波 (1983-), 男, 广西南宁人, 硕士研究生, 研究方向为嵌入式系统; 王 健, 副教授, 研究方向为计算机体系结构、嵌入式系统; Rusty Lynch, Intel 开源技术中心软件工程师, 研究方向为嵌入式 Linux。

1.2 目标

经以上分析,如果能够设计一个播放器,其体系结构具备 gstreamer 的开放性,同时又能够像 helix-player 一样支持丰富的 mime type 类型的媒体文件^[5],这将为 Linux 赢得更多用户,从而促进 Linux 更加稳步向前发展。这也是文中的目的所在。

2 设计新一代更加开放体系结构的播放器

2.1 新体系架构的播放器的先进性

首先分析现有 Linux 系统上的传统播放器,基本上所有的播放器其 GUI 部分和真正的播放部分都是形影相随,不分你我的,存在很高的耦合,没有 GUI,也就没有下面的用户播放了。而如果把这两部分分离开来,会有什么好处呢?文中所设计的播放器的重点就是把这两部分分离开来,降低甚至消除这两部分的耦合性,消除耦合有很多的优势,这样 OSV 可以更加专注于设计自己风格的 GUI,而支持解码的如 realnetwork 则能够更加专注于支持更多的 mime type。虽然把这两部分分开了,但是从用户的角度而言,用户依旧是点击某个按钮,播放器响应用户的播放请求进入播放状态。这样就需要一个信息载体来承载这个点击事件,使得这个 GUI 的点击事件能够被播放部分^[6](俗称 engine,下文将以 engine 来进行阐述)捕获。这个载体就是现在 Linux 系统里面占据很重要部分的 Dbus。顾名思义,DBus 就是一个 bus,即总线,但不是硬件体系结构里面的总线,而是纯粹的软件系统里面的总线,进程间的通讯都由其来完成^[7]。

2.2 新一代播放器的体系结构

图1是文中所设计播放器的体系架构,从图中可以看出,最下层是 engine,最上层是 GUI,用户的播放命令最终会传到 engine,然后由 engine 处理播放请求,DBus 负责传达这个命令,每个 engine 都是一个独立的进程,而其各自的 XXXX-dbus-server 也是一个独立的进程。

2.3 功能模块分析

(1)用户:负责发出播放拖曳等命令。

(2)GUI:负责与用户交互收集用户命令,同是分析被请求播放文件的 mime type,把这些信息发送给下一级模块。

(3)Engine Management:负责管理底层的 engine,对用户的每次的播放请求,根据每个播放器的特长做

出合适的决策,选择合适的 engine 来支持用户的播放请求。

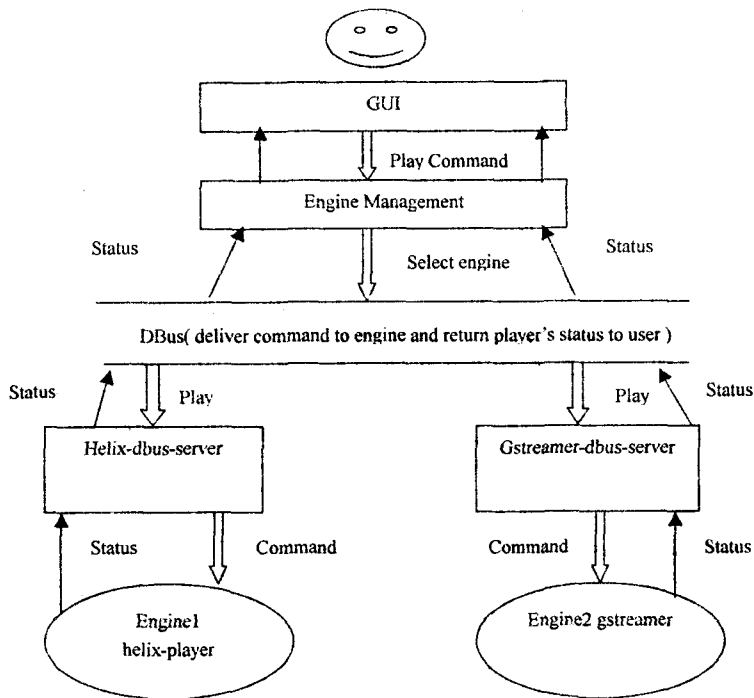


图1 新播放器体系结构

(4)DBus: 该模块是 Linux 系统中存在的,目的是统一进程间的通讯,使得进程间通讯都通过 Dbus 这个软件总线。Engine Management 模块做好决策以后,把命令发送到总线上,此时已经决定好该选择哪个 engine 来播放,总线负责将请求送到被选择的 engine 相对应的 XXXX-dbus-server 上。

(5)XXXX-dbus-server(XXXX=helix,gstreamer):这一层是介于 Dbus 和 engine 之间,主要是负责对外暴露 engine 的接口给 Dbus,使得 Dbus 可以通过这些接口控制 engine。

(6)Engine N(n=1,2): 根据收到的播放命令,做出响应的动作,同时更新状态(status)并返回该信息返回给 GUI,增加可用性和友好性。

3 流程和代码分析

3.1 主要场景

结合图1,分析播放器典型的三个播放场景:

场景一,用户通过 GUI 选择播放文件 a. rmvb,然后触发播放事件,调用 Engine management 模块的选择 engine 方法,该方法通过查询本地数据库得知 helix-player 支持 rmvb,因此,将播放命令通过 Dbus 发送给在 Dbus 另一端等待的 helix-dbus-server, helix-dbus-server 收到命令以后再传达给 helix-player engine, helix-player 处理 a. rmvb,没有出错,将状态信息按原

路返回给 GUI 以更新 GUI 状态。若出错,提示用户文件处理出错。

场景二,用户选择 a. mp3, 由于 helix - player 和 gstreamer 都支持 mp3,此时 Engine management 的决策就可以二者任选一了。

场景三,若用户的播放请求文件在 Engine management 的本地数据库都得不到支持,此时将 Engine management 所持有的 engine 列表 engineList 都试一遍,如果某个 engine 成功播放了该文件,则更新该 engine 对应的数据库,进入播放状态,如果都得不到支持,则提示用户播放失败。

3.2 代码分析

除了两个 engine 和 helix - dbus - server 模块,其余各部分都是用 python 实现的,关于 python 的介绍不在文中的阐述范围,由于项目庞大,不可能一一详述,而实现更加开放体系结构的播放器,关键是跟 Dbus 相联系的那两层,分别是 XXXX - dbus - server 和 engine management 层,而 GUI 可以根据 OSV 甚至用户的风格自己定义设计。

(1)helix - dubs - server:这一层主要是负责跟 engine 交互,起到承上启下的作用,首先该模块必须向 Dbus 注册服务,将 play, pause, seek, stop 等播放器的接口暴露给 Dbus 以供上层调用实现媒体播放。

(2)gstreamer - dbus - server:该模块跟 helix - dbus - server 功能是一样的,都是向 Dbus 注册以负责暴露接口,不一样的是它是基于 python,而 helix - dbus - server 是基于 c。

(3)engine management:该模块在这个体系结构中的重要程度不言而喻,用户的播放命令首先会在这里汇集,而这又是通往 engine 的唯一途径,因此最重要的数据结构就是 engineList,保存 Dbus 上可用的 engine 的接口,同时也需要通过 Dbus 跟 engine 交互,目的是得到每个 engine 的播放能力,以便在做出选择决策的时候提供参考,engine 的播放能力是 engine management 模块最急需也是最有价值的信息,因此每个 engine 都应该通过各自的 XXXX - dbus - server 向 Dbus 暴露一个接口,不妨称之为 getCapability, Engine management 通过垂询该接口,就能知道每个 engine 的具体情况了。

由于代码量大,不能一一枚举,但文中相关的所有代码都可以参考 <http://www.moblin.org/repos/>,其上有整个项目的源代码。

4 结束语

设计了基于 Intel 08 平台 MIDLinux 的新一代媒体播放器,在降低播放器模块之间耦合度的同时,将 helix - player 和 gstreamer 两个当前流行的 Linux 播放器合二为一,为 Intel 08 平台移动设备未来的软件系统提供了强大的媒体播放能力。

Linux 开源软件虽然众多,但都不是很完善,DBus 的出现使得 Linux 桌面系统进程间的通讯变得更加简单容易^[8],也使得 Linux 开发者在开发设计软件时多了一种选择,那就是以 Dbus 为信息媒介降低软件模块之间的耦合度,提高软件的可扩展性,从而提高软件的质量。从文中的角度,未来不管是已经存在的播放器,还是正在诞生的播放器,支持 Dbus 将成为一种趋势。

参考文献:

- [1] GStreamer Open Source Media Framework[EB/OL]. [2003 - 07 - 03]. <http://people.debian.org.tw/chihchun/2003/07/03/gstreamer-open-source-media-framework>.
- [2] Comparison of media players[EB/OL]. [2008 - 08 - 27]. <http://en.wikipedia.org/wiki/Comparison-of-media-players>.
- [3] Helix DNA Server Process Model[EB/OL]. [2008 - 02 - 25]. <https://helix-server.helixcommunity.org/2003/devdocs/processmodel.html>.
- [4] GStreamer Application Development Manual [EB/OL]. [2008]. <http://gstreamer.freedesktop.org/data/doc/gstreamer/head/manual/html/index.html>.
- [5] Helix DNA Client Architecture[EB/OL]. [2003 - 02 - 25]. <https://client.helixcommunity.org/2003/devdocs/intro/client-intro.htm>.
- [6] D - BUS based multimedia server[EB/OL]. [2003 - 03 - 12]. <http://osdir.com/ml/video.gstreamer.devel/2003-03/msg00031.html>.
- [7] D - Bus Tutorial[EB/OL]. [2007 - 08 - 29]. <http://dbus.freedesktop.org/doc/dbus-tutorial.html>.
- [8] The xine hacker's guide[EB/OL]. [2000 - 07]. <http://xine-hq.de/index.php/hackersguide>.

(上接第 201 页)

安电子科技大学出版社,2006.

[3] OPNET Technologies, Inc. OPNET Modeler PLA3 在线文档[EB/OL]. 2002. <http://www.opnet.com>.

[4] Banks J, John C. Discrete - event System Simulation[M]. 3rd

ed. [s.l.]: Prentice Hall, 2001.

[5] 刘晓静, 苏锦海. OPNET 在网络规划设计中的应用[J]. 微机信息, 2006, 22(2): 104 - 106.