

互动：数据分析与云计算应用案例（征集|参与）

rockmylife

暂无签名

首页 | 博文目录 | 关于我



moses731

博客访问：230515

博文数量：163

博客积分：4045

博客等级：上校

技术积分：1200

用 户 组：普通用户

注册时间：2007-06-15 12:56

加关注

短消息

论坛

加好友

文章分类

- 全部博文（163）
- 职业人生（2）
- 无线技术（5）
- Arm（6）
- Test（4）
- shell（19）

转载 C语言实现插件机制 2013-03-07 16:50:41

分类： 嵌入式  
原文地址：C语言实现插件机制 作者：mmmmpl

现在越来越多的软件项目都提供插件机制，这样使得软件的扩展性大大增强，那么到底插件机制的实现是怎么样的呢？在这里只谈论C语言的实现，其实C语言实现插件的例子也很多，像mjpg-streamer就是将输入输出做成插件，dm500机顶盒的主程序enigma也使用了插件机制，我就是从enigma当中学习的。好了，这里给一个简单的例子来看看如何实现。

我的设想是这样的：有一个主程序，有一个插件(.so文件),主程序里面提供一些最基本的功能模块，而在插件中使用这些功能模块实现某些功能。简单点就是主程序要调用.so文件中的函数，而.so要调用主程序中的某些基础函数。

为了既能使用插件a.so, 又能使用插件b.so, 还能使用插件c.so,这里要显式调用动态链接库，即通过dlopen, dlsym, dlclose来使用.so文件，还是直接来看代码。

```
#mkdir soplugin
#cd soplugin
#vim main.h
```

```
//main.h
```

```
#ifndef MAIN_H
#define MAIN_H
```

```
class A {
public:
    A();
    ~A();

    void p(const char *s);
};
```

- Perl ( 0 )
- Python ( 1 )
- Php ( 0 )
- Network ( 8 )
- Linux ( 102 )
- English ( 1 )
- C/C++ ( 14 )
- 未分配的博文 ( 1 )

文章存档

- 2013年 ( 34 )
- 2012年 ( 1 )
- 2010年 ( 12 )
- 2009年 ( 8 )
- 2008年 ( 83 )
- 2007年 ( 25 )

我的朋友

最近访客



cu\_ggg



daimadu



Windeal



zjpixyni



飞机塞进



a663200



雨笑儿



hwayw



zhangjin

微信关注



IT168企业级官微  
微信号: IT168qiye

```
#endif

#vim main.cpp

#include
#include // dlopen/dlsym/dlclose头文件

#include "main.h"


void A::p(const char *s)
{
    printf("A::p()\n");
}

void plugin_show(const char *s) // 这个函数会被.so文件调用
{
    printf("%s\n", s);
}

int main(int argc, char *argv[])
{
    int (*PluginExec)(int argc, char *argv);
    void *plugin;


    printf("loading...\n");
    plugin = dlopen("./tplugin.so", RTLD_GLOBAL|RTLD_NOW); // 显式打开.so文件
    if (plugin == NULL) {
        printf("ptr: %p\n", plugin);
        perror("Can not load tplugin.so");
        return -1;
    }


    PluginExec = (int (*)(int, char*))dlsym(plugin, "plugin_exec"); // 得到入口函数指针
    if (PluginExec) {
        PluginExec(0, NULL); // 调用入口函数
    }
```



系统架构师大会  
微信号：SACC2013

订阅

 订阅

 订阅到 Google

推荐博文

·lesson3-Qt对话框

·BLE-NRF51822教程3-sdk程序框...

·Greenplum(GPDB)开源啦！~...

·生命不息，折腾不止

·PostgreSQL不同模式(SCHEMA)...

·sqlplus scott/system@ora10g...

·使用oradebug dump hanganaly...

·oracle em 修改端口

·使用oradebug dump hanganaly...

·使用oradebug dump hanganaly...

热词专题

·Eclipse-sdcc学习（ fedoar 14...

·Linux 环境定制：我的 .bashr...

·Android游戏开发设计步骤...

·nagios+cacti整合（ 中文版 ）...

·服务器IIS6的PHP最佳配置方法...

```
dlclose(plugin);

return 0;
}

#vim plugin.cpp
#include "main.h"

extern void plugin_show(const char *s);

extern "C"
int plugin_exec(int argc, char *argv[]) // 这里要用extern "C"声明，否则C++编译器会给函数名加上一些乱七八糟的东西，不信你可以试试，然后用objdump去查看
{
    A a;

    a.p("in plugin_exec"); // 这个在主程序中实现

    plugin_show("in plugin."); // 这个也在主程序中实现

    return 0;
}

好了，代码就这些，很简单，但能说明问题就行了，再写个Makefile。

# vim Makefile
all:
    g++ -shared -fPIC -DPIC -c plugin.cpp -l.
    ld -shared -o tplugin.so plugin.o
    g++ -Wl,-E -o mm main.cpp -ldl

    前面两行是将plugin.cpp做成一个.so文件，后面一行是编译main.cpp，这里要特别注意参数-Wl,-E，这个参数意思是将-E参数传递给链接器ld,最终的目的是将main.cpp中的函数输出成全局符号，以方便.so文件调用，如果没有此参数，那么编译也不会有问题，但在运行时dlopen总是会失败，原因是无法解决符号依赖问题。关于这个参数你可以用objdump对比一下加与不加的结果差别。

    好了，接下来编译然后运行。

#make
```

```
g++ -shared -fPIC -DPIC -c plugin.cpp -l.
ld -shared -o tplugin.so plugin.o
g++ -Wl,-E -o mm main.cpp -ldl
./mm
loading...
A::p()
in plugin.
```

这个例子只是简单的实现打印，但至少已经可以看到主程序和.so文件之间可以调用了，那我再实现a.so， b.so当然也不成问题了。可能有人会产生疑问，为什么不隐式调用呢？原因是：如果隐式调用就必须在编译阶段确定好.so文件，这样就谈不上可扩展插件了，它们之间就存在了编译上的依赖关系。而显式调用是在运行期间确定他们的依赖关系的。

如果有兴趣可以参数mjpg-streamer去学习，但是mjpg-streamer中的插件没有调用主程序的函数，最好的学习例子还是enigma,他里面实现了大量的插件。以后嵌入式软件项目的扩展性要求会越来越高，插件扩展也大受欢迎，插件扩展的机制很多，需要我们去收集学习，这里讲的是最简单的一种。我为什么要学习，原因是我目前参与的一个项目主程序尽然大到30多MB，编译链接时间太长，要扩展功能就更痛苦了，一次一次的编译/链接，一次一次的等待，真是折磨。

附：

enigma是dm500机顶盒的主程序，enigma本身通过c++实现gui, gdi, dvb等一堆基础库，并实现了插件管理器，外围的功能基本全是c/c++插件实现。  
enigma2是dm800的机顶盒主程序，enigma2就比enigma更高级了，他通过c++实现gui, gdi, dvb等一些基础库，其余的界面功能，机顶盒功能全部是用动态语言python实现，中间使用swig胶合在一起。也是不错的学习例子，但项目太大，学习不太容易。



阅读(33) | 评论(0) | 转发(0) |

上一篇： 802.11n无线网络技术剖析  
下一篇： ARM学习之三:串口的配置

0

赞

相关热门文章

热

RHCS原理及操作

热

lesson2-信号与槽机制

热

openwrt中使用ubus实现进程通...

热

Linux 下svn恢复到某一版本...

热

CAS实现SSO单点登录原理...

荐

SHTML是什么\_SSI有什么用...

荐

shell中字符串操作

荐

shell中的特殊字符

荐

关于java中的“错误：找不到或...

荐

linux设备驱动归纳总结...

热

linux dhcp peizhi roc

热

关于Unix文件的软链接

热

求教这个命令什么意思，我是新...

热

sed -e "/grep/d" 是什么意思...

热

谁能够帮我解决LINUX 2.6 10...

给主人留下些什么吧！~~

评论热议

请登录评论。

[登录](#) [注册](#)

## 艺术漆

## 女款毛线帽子

宝宝帽子图片

## 简约中式装修

ee留学

## C++程序设计

## 小户型简约装修

艺术漆

艺术漆推广

[关于我们](#) | [关于IT168](#) | [联系方式](#) | [广告合作](#) | [法律声明](#) | [免费注册](#)

Copyright 2001-2010 ChinaUnix.net All Rights Reserved 北京皓辰网域网络信息技术有限公司. 版权所有

感谢所有关心和支持过ChinaUnix的朋友们

京ICP证041476号 京ICP证060528号