插件安装方案

1. **方案选择**

目前常见的套件管理方案，类[Debian](http://en.wikipedia.org/wiki/Debian)系统（比如ubuntu）用的是dpdk，类red hat系统（比如centos、fedora）用的是rpm， 这些方案在嵌入式中会显得有些臃肿。嵌入式系统中的套件管理可选用opkg方案，openwrt与极路由（也是基于openwrt）的插件安装方式都是使用此方案。

1. **opkg方案解析**

* Ipk插件包格式

使用ipkg-utils套件中的ipk-build命将插件打包成.ipk包，包含内容如下：

* + control文件

这个文件包含着插件包的基础信息，名称、版本、依赖库的关系、基于何种cpu架构等信息。

* + data.gz

这个gz包压缩着插件包的实体内容，通常会把这个包从根目录解压到目标系统中，运行命令、库文件、配置文件都会相应这放到指定的目录下。

将来我们的插件通常会安装到tf卡中。

* + preinst,postinst,prerm,postrm

这四个脚本分别代表安装前执行，安装后配置，删除前执行，删除后配置，满足每个插件安装卸载前后的自定义操作。

* 管理ipk插件包

使用opkg命令管理ipk软件包。Opkg具有的常见命令操作如下：

* + Install 安装插件

有三种方式安装

1. 指定网络插件包路劲。
2. 下载到本地的软件包
3. 使用插件别名，使用插件别名时需要首先update将当前系统可用插件的内容更新到目标系统。

如果安装插件之前，使用了update操作，会将插件包信息放到目标系统中，安装过程中会校验插件包的md5值，如果不匹配是不会安装成功。反之，则不会检测，会直接安装。

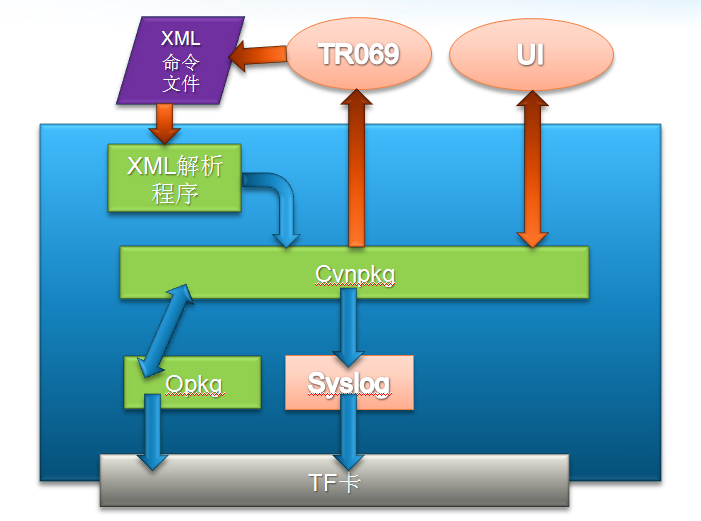
这样我们必须做的处理就是，定时update同步插件包信息下来，并且在程序中判断，如果没有插件包信息的话，则不予安装。

* + Remove 卸载插件
  + Update 更新当前系统的可用插件

/etc/opkg.conf配置文件制定一个url，使用该url来更新当前系统可用插件。

* + List 列出当前系统可用插件
  + List-installed 列出当前系统已安装插件
  + Download 下载插件到当前目录

1. **插件系统平台架构**
   1. **整体构架图**



架构图

* 1. **构架图说明**

1、套件管理程序提供一组公开的接口，我们可以通过设备服务器tr069协议（系统级插件），或者web页面（用户级别插件）对插件进行管理。并且管理程序会将操作记录保存在日志文件中。

2、套件管理程序可以分成三个部分：XML解析程序、cvnpkg、opkg。

1. XML解析程序：提供一种批量化管理插件的方式。
2. cvnpkg：opkg的封装，附加了日志记录功能。
3. opkg：提供安装、卸载、更新、查询等基本功能。
4. **系统级插件管理**
   1. **流程图**



流程图1

* 1. **安装流程说明**
     1. **下载安装流程**

1、设备开机即通过tr069协议，注册到ACS设备服务器上。

2、设备服务器下发插件升级命令，升级命令中包含着一个url地址，指向一个包含升级操作的文件。

3、tr069客户端获取升级操作文件。

4、触发套件管理程序升级操作。

第1~4步，为tr069协议的标准升级流程。

5、解析升级操作文件，文件以XML的方式组织，每个节点代表一个软件的操作，可能是安装，走步骤6；也有可能是卸载，走步骤7。

6、XML节点中包含插件安装包的URL，套件管理程序根据此URL下载插件包。

7、根据XML节点中的操作，执行安装或卸载流程。

8、上报安装结果给设备服务器。

第8步上报结果是考虑到，终端的操作都是被动型的，设备服务器在下发命令之后，希望获得结果信息。但其实目前在我们的tr069协议中是没有这种事件上报机制的，所以该步骤属于待定。

* + 1. **信息上报流程**

9、设备服务器下发查询命令，比如查询已安装插件列表。

10、触发套件管理程序查询操作。

第9~10步，为tr069协议的标准查询流程。

11、返回查询信息给设备服务器。

* 1. **系统固件更新后处理**

1. **云网管配置系统级插件流程**
   1. **概述**

插件的配置可以分为两种：

1. 可被云网管管理，属于基础配置，存放于公共配置库，这是我们需要讨论的配置。
2. 不被云网管管理，在插件安装时，即被postinst脚本初始化的配置。
   1. **流程图**



流程图2

* 1. **配置流程说明**

1、设备开机即通过tr069协议，注册到ACS设备服务器上。

2、设备服务器下发更新配置命令，命令中包含着一个url地址，指向文件服务器存放的配置文件。

3、tr069客户端获取配置文件。

4、触发公共配置模块配置更新操作。

第1~4步，为tr069协议的标准流程。

5、解析配置文件，更新配置。

6、获取已安装插件列表，查询插件信息。

7、重启插件，加载新的配置。

1. **用户级插件管理**
   1. **概述**

用户级插件属于用户可以自行安装，卸载，更新的插件。区别于之前提到的插件，为系统级插件，无法被用户感知，只受设备服务器管理。

用户级插件另一个重大特点是，配置可被用户修改，并且在更新插件的时候，配置必须得以保存。

* 1. **安装流程**
     1. **流程图**



流程图3

* + 1. **流程说明**

1. 用户界面点击插件进行安装，调用用户插件接口。
2. 从文件服务器下载插件包。
3. 安装插件。
4. 返回安装结果。
   1. **卸载流程**
      1. **流程图**



流程图4

* + 1. **流程说明**

1、用户界面点击插件进行安装，调用用户插件接口。

2、卸载插件。

3、返回卸载结果。

* 1. **更新插件**
     1. **流程图**



流程图5

* + 1. **流程说明**

1、用户界面点击更新插件按钮，调用用户插件接口。

2、下载插件包。

3、更新插件包，保存原有配有配置。

4、返回更新结果。

* 1. **查询插件**
     1. **流程图**



流程图6

* + 1. **流程说明**

1、用户界面点击插件进行安装，调用用户插件接口。

2、查询插件信息。

3、返回查询信息。

* 1. **插件配置**

用户插件配置是没有类似公共配置模块来集中管理的，有些插件是不需要进行配置的，比如诸多的加速插件，可配置的插件则需要用户界面来呈现，并且考虑到插件本身的独立性与可移植性，可为每个插件配套一个配置修改程序。

* + 1. **流程图**



流程图7

* + 1. **配置读取流程说明**

1、用户界面会调用插件配置程序查询接口。

2、插件配置程序从配置文件读取配置。

3、返回给用户界面展现配置项。

* + 1. **配置修改流程说明**

4、用户界面调用插件配置程序设置接口。

5、插件配置程序从修改配置到配置文件。

1. **接口定义**
   1. **套件管理程序插件接口**

参考架构图可知，套件管理程序定义的接口有两个

1. 批量处理接口：

输入：XML文件，一组插件操作集合。

处理：解析XML文件，获取插件操作指令与指令参数，进行相应安装或卸载。

输出： 执行结果。

1. cvnpkg管理接口

输入：cvnpkg 命令[ install|uninstall|upgrade|info ]

处理：调用opkg，执行查询命令。

输出：返回查询结果。

* 1. **批处理XML文件格式定义**
     1. **格式如下**

<pluginstore>

<plugin command="INSTALL">

<name>portal</name>

<url>http://.....</url>

<urgent>true</urgent>

<md5>334123CE4527A48E30CC37CD57E99CF9</md5>

<hardware-version>3.4.3</hardware-version>

</plugin>

<plugin command="UNINSTALL">

<name>lighttpd</name>

<software-version>1.2.3</sofeware-version>

<hardware-version>3.4.3</hardware-version>

</plugin>

</pluginstore>

* + 1. **格式解析**

**Pluginstore**代表着整个插件操作列表。

**Plugin**代表单个插件操作，属性**command**定义了两种操作，安装INSTALL与卸载UNINSTALL。

**Name**代表插件名称。

**url**代表从该插件存于服务器的网络路劲，UNSTALL操作忽略此选项。

**Urgent**代表插件的紧急程序，如果为true，代表紧急，为flase则非紧急。

**Md5**下载插件后，校验md5值，如果不相同，则拒绝安装，UNINSTALL操作忽略此选项。

**Software-version**插件包软件版本，如果版本不符，拒绝卸载命令，INSTALL操作忽略此选项

**Hardware-version**固件版本号，如果版本不符，拒绝安装和卸载。

* 1. **Cvnpkg接口指令**
     1. **指令格式**

cvnpkg 命令[ install|uninstall|upgrade|info|list-installed ]

* + 1. **指令定义**

|  |  |
| --- | --- |
| 命令选项 | 描述 |
| list-installed | 查询已安装插件列表 |
| Info | 查询单个安装插件信息 |
| install | 安装插件 |
| uninstall | 卸载插件 |
| upgrade | 更新插件 |