第 33 卷 第 21 期 计 算 机 工 程 2007 年 11 月
Vol.33 No.21 Computer Engineering November 2007
·软件技术与数据库· 文章编号：1000—3428(2007)21—0061—02 文献标识码：A 中图分类号：TP311.5

基于 ipkg 包管理程序嵌入式 Linux 的构建与维护

宋劲杉，袁 涛
(清华大学自动化系汽车电子研究中心，北京 100084)

摘 要：ipkg 包管理程序使开发者无需繁琐的交叉编译从而快速构建系统，并使软件的升级过程自动化。该文介绍了 ipkg 包管理的概念和软件包的格式，说明了如何从一个空根文件系统开始建立 ipkg 软件环境（即 bootstrap 过程），以嵌入式 Web 服务器为例介绍了构建应用系统的过程，并以某 Qt 图形系统为例介绍软件包的制作过程。

关键词：ipkg；ARM；嵌入式 Linux

Construction and Maintenance of Embedded Linux Based on Ipkg Package Management Program

SONG Jin-shan, YUAN Tao
(Automobile Electronics Research Center, Department of Automation, Tsinghua University, Beijing 100084)

【Abstract】 With ipkg package management program, developers can avoid trivial steps involving cross compilation, quickly build up embedded Linux system, and enable the software to upgrade automatically. This paper introduces the concept of ipkg package management and the package format, and presents the bootstrap process. Taking embedded Web server as an example, the construction of an application system is illustracted. And a package construction process with a Qt graphic system is introduced.

【Key words】 ipkg; ARM; embedded Linux

Debian Linux 的成功源于其独特而完备的体系结构，安装软件包时只需一条 apt-get 命令即可自动检测依赖关系并自动下载安装该软件包依赖的其他软件包，避免了配置、编译、安装、解决依赖缺失和依赖冲突等许多繁琐的问题，这一切得益于 dpkg 包管理程序。但嵌入式 Linux 领域中始终没有一个完善的包管理程序，开发者仍然按 configure/make/make install 的步骤安装软件包，然后手动解决各种依赖问题。

现在，致力于为 iPAQ 和 Zaurus 掌上电脑定制嵌入式 Linux 系统的开源项目 Familiar 和 OpenZaurus 开发出完善的 ipkg 包管理系统，开源社区随之发布了丰富的软件包。虽然这些软件包是针对 XScale ARM 交叉编译的，但只有一些依赖于板级硬件，大部分可用于其他 ARM 处理器，例如 S3C2410/2440、AT91RM9200 等。在这些平台上安装软件如同在 PC 的 Debian Linux 上一样方便。

1 ipkg 包管理程序简介

ipkg 是针对存储器容量不大的嵌入式 Linux 设计的包管理程序，与 Debian Linux 的 dpkg 相比非常小巧，如表 1 所示，数据摘自文献[1]。

表 1 ipkg 与 dpkg 的比较

比较项目	ipkg	dpkg
包管理程序大小	约 13KB	几百 KB
软件包元数据（名称、体系结构、依赖、文件列表、安装和删除配置脚本等）大小	以 iPAQ 上的 Familiar 发行版为例，16MB 的压缩 Flash 映像中约有 38KB 元数据	典型值几 MB
软件包的细分程度	非常细，用户可以只安装程序、共享库等必要文件而不安装文档、头文件、静态库等无用文件	比较粗，许多软件包中包含文档文件

此外，ipkg 还有以下优点：

(1)网络上有大量可自由使用的软件包，开发者无须进行繁琐的交叉编译，只须专注于软件的配置使用。许多软件（如 Python 和 Apache）很难交叉编译，由于其编译过程往往分成几个阶段，后一阶段需要运行前一阶段编译的程序，而交叉编译的程序由于无法在主机上运行，因此后一阶段会失败。现在有许多交叉编译成功的软件包发布，开发者不必再重复这些工作。

(2)ipkg 拒绝安装不满足依赖关系的软件包，因此，可以有效地管理依赖问题。而手工编译安装的方法却要面对各种问题，比如某软件所依赖的共享库没有安装，运行时会提示找不到共享库，这种错误相对容易判断；但如果已安装了一个低版本的共享库，该软件运行时不会提示找不到共享库，而是出现各种奇怪的错误，这就很难判断原因了。

(3)ipkg 使删除和升级软件包变得十分便捷。每个软件都由分布于 bin, lib, etc 等目录的文件组成，手工删除文件或升级版本繁琐且容易出错，ipkg 根据软件包元数据将这一过程自动化。为便于开发时项目管理和用户使用时自行升级，开发者应把手工编译安装的软件也制作成软件包。为方便用户升级，ipkg 还配有图形界面的前端程序（如 Qt 的 qipkg）。

但 ipkg 也有不足之处：

(1)网络上可用的软件包大部分是用 arm-linux-gcc 编译的，针对 ARM 平台和 glibc 库，如果采用 MIPS 等其他平台或者在 ARM 平台采用更精简的 uclibc 库则很难找到相应的软件包。但 ipkg 程序本身可以交叉编译到其他平台，即使没有丰富的软件包可用，ipkg 仍可以在开发和维护时用于软件

作者简介：宋劲杉(1982－)，男，硕士研究生，主研方向：嵌入式系统及应用；袁 涛，副教授

收稿日期：2006-12-26 **E-mail：**songjinshan@gmail.com

的打包管理。

(2)ipkg 目前缺少 Debian Linux 那样统一的软件仓库，比如 Familiar 和 OpenZaurus 项目都有各自的软件包划分体系和依赖关系树，属于不同项目的软件包不能通用。

这些有待于 ipkg 体系的进一步完善和规范才能解决。

2 ipkg 软件包格式及 bootstrap 过程

ipkg 软件包以.ipk 为文件名后缀，由 control.tar.gz, data.tar.gz, debian-binary 3 个文件以 ar 或 tar 命令打包而成。其中，data.tar.gz 包含程序文件及路径信息；control.tar.gz 中的 control 文件保存软件包元数据，也可能包含 conffiles, preinst, postinst, prerm, postrm 等脚本文件，在安装前后和删除前后执行适当的配置；debian-binary 文件保存软件包格式的版本号。

用 ipkg 程序构建系统就是从一个空根文件系统开始逐步把所有安装的软件纳入 ipkg 管理的过程。首先需要安装 ipkg 程序并建立 ipkg 的运行环境，这个过程称为 bootstrap。具体步骤如下：

(1)参照文献[2]第 6 章，在主机上建立/bin, /sbin, /dev, /lib, /usr 等基本目录，用 mknod 命令建立 /dev/console, /dev/ttySAC0, /dev/tty, /dev/null 等设备节点。

(2)从 Familiar 的网站(http://familiar.handhelds.org)上下载 busybox, libc, libipkg, ipkg 等软件包，运行 ar x 或 tar zxvf 命令将程序文件解包至相应目录。手工安装阶段没有自动检查依赖的机制，因此，必须额外注意依赖关系（如 ipkg 依赖于 libipkg0 和 libc6）。

(3)为 busybox 配置 shell 运行环境。参照文献[2]第 6 章，配置/etc/inittab, /etc/init.d/rcS, /etc/profile 等启动脚本，创建账号文件/etc/passwd, /etc/group，创建 mount 命令配置文件 /etc/fstab。

(4)将根文件系统烧写到目标板或以 NFS 方式挂载，在目标板上启动 shell，运行 ipkg install 命令，将先前手工安装的软件包覆盖安装一遍。ipkg 将软件包元数据记录在/usr/lib/ipkg/目录并运行 preinst 和 postinst 配置脚本。

3 ipkg 构建应用系统的实例

thttpd和Goa都是设计小巧、适用于嵌入式系统的Web服务器，且都支持CGI。CGI可以直接用C语言写，或用perl, Python等解释程序，还可以用php等网页内嵌的服务器脚本。以上所有方案都有可用的软件包，经过简单的安装和配置即可使用。这里以httpd和Python CGI为例[3]。

经过了上述 bootstrap 步骤之后，安装 thttpd 和 Python 解释器只需要几条命令：

```
ipkg install thttpd_2.25b-r0_arm.ipk
ipkg install libpython2.4-1.0_2.4.1-ml5_arm.ipk
ipkg install python-core_2.4.1-ml2_arm.ipk
```

安装之后对 thttpd 进行配置，创建/etc/thttpd.conf，内容如下：

```
dir=/srv/www
# chroot
user=root
logfile=/dev/null
pidfile=/dev/null
cgipat=/cgi-bin/*
```

注意：不要设置 chroot，否则运行 CGI 程序时找不到 python 解释器。最后在/etc/inittab 中添加：

```
::respawn:/usr/sbin/thttpd -C /etc/thttpd.conf
```

编写一个简单的 CGI 测试程序：/srv/www/cgi-bin/hello.py，内容如下：

```
#! /usr/bin/python
print "<html><body>Hello World!</body></html>"
```

注意：hello.py 必须设置可执行位才能作为 CGI 程序执行。之后就可以从主机打开 Web 浏览器查看该页面。

ipkg 软件包划分得很细，Python 被分成许多软件包，安装了 python-core 之后只有解释器和最基本的 module，若需要其他 module 则要安装相应的软件包，比如正则表达式需要 python-re 包，Qt 程序需要 python-sip 和 python-pyqt 包。

4 ipkg 软件包制作实例

Familiar项目有 2 套Qt图形系统[4]，其中，qpe的版本太旧；opie的底层软件包依赖iPAQ硬件（触摸屏、蓝牙等），在S3C2410 平台上不能正常运行。因此，笔者下载了Qt源码自己交叉编译并做成软件包以便升级维护。虽然可以用tar zcf或arr命令按上述ipkg软件包格式打包，但用ipkg-build脚本（由ipkg-utils包提供）打包更为可靠，它会自动检查开发者提供元数据的完整性，如果缺少某些必需字段将提示相应的错误。

首先创建一个用于打包的目录，按照根文件系统中的目录层次将待安装文件复制到打包目录，本文将所有文件安装到 opt/QtPalmtop 下 然后在打包目录下创建 CONTROL 目录，在 CONTROL 目录下创建 control 文件记录软件包元数据：

```
Package: myqpe
Version: 1.0
Architecture: arm
Maintainer: SONG Jin-Shan
Section: graphics
Priority: optional
Description: Qt/Embedded & Qtopia
Source: qtopia-free-source-2.1.1.tar.gz, qt-x11-2.3.2.tar.gz,
qt-embedded-2.3.10-free.tar.gz
Depends: libc6, e2fsprogs-libuuid1, libjpeg62
```

将交叉编译时用到的源码包记录在 Source 字段中有助于软件包的使用者了解其内部组件的版本，在出现 Bug 时有助于获取解决问题的信息。交叉编译时用到的 libuuid 和 libjpeg 共享库应在 Depends 字段指明，因此，ipkg 要求目标系统先安装 Depends 的软件包再安装本软件包。然后在 CONTROL 目录下创建 postinst 安装后配置脚本：

```
#! /bin/sh
exist=`sed -n "/\/opt\/QtPalmtop\/lib/p" /etc/ld.so.conf`
if [ -z "$exist" ]; then
echo -e "\n/opt/QtPalmtop/lib" >> /etc/ld.so.conf
fi
ldconfig
exist=`sed -n "/\/opt\/QtPalmtop/p" /etc/profile`
if [ -z "$exist" ]; then
echo -e '\nexport QTDIR=/opt/QtPalmtop' >> /etc/profile
echo 'export QPEDIR=/opt/QtPalmtop' >> /etc/profile
echo 'export PATH=$PATH:/opt/QtPalmtop/bin' >> /etc/profile
echo 'export QWS_MOUSE_PROTO=USB:/dev/input/mice' >>
/etc/profile
fi
```

注意：postinst 必须设置可执行位。该脚本的作用是更新 /etc/ld. so.cache 加入新的共享库，并在启动脚本中设置 Qt 运行所需的环境变量，最后用 ipkg-build 脚本对整个目录打包生成 myqpe.ipk。

（下转第 65 页）