基于 Yocto 订制嵌入式 Linux 发行版

杜登科，胡爱兰，李林峰，张玉生
（华北计算机系统工程研究所，北京 100083）

摘　要：随着工业科技的发展，嵌入式 Linux 展现了巨大的价值，广泛应用于消费电子设备、网络设备和工控机领域。广泛的应用带来了各种各样的需求，各大厂商纷纷订制自己的嵌入式 Linux 产品，导致大量的重复工作，规范和标准差异化越来越大。在开放源代码的精英发起下，Yocto 横空出世，以开源协作的方式逐步统一嵌入式 Linux 的订制和开发标准，避免许多重复工作，极大地简化了嵌入式 Linux 系统的订制，并且省掉大量的重复工作，使嵌入式 Linux 应用开发者可以将重心放在自己的工作上。

关键词：嵌入式；Linux；Yocto；开放源代码
中图分类号：TP368.2　　**文献标识码**：A　　DOI：10.19358/j.issn.1674-7720.2016.14.022
引用格式：杜登科，胡爱兰，李林峰，等. 基于 Yocto 订制嵌入式 Linux 发行版[J]. 微型机与应用，2016，35(14)：68-70.

Customizing Linux distribution based on the Yocto

Du Dengke, Hu Ailan, Li Linfeng, Zhang Yusheng
(National Computer System Engineering Research Institute of China, Beijing 100083, China)

Abstract: With the development of society, the embedded Linux demonstrates the great value, and it is widely used in consumer electronics devices, network equipment and IPC fields. Wide application brings various demond. The major manufacturers have customized their own embedded Linux products, which produced much duplication work, and the specification and standard's differentiation got bigger. Under the guid of open source's elites, the Yocto project was born. The aim of Yocto project was integrating the embedded Linux's development and standard specification and avoiding the duplication of works in the development. The Yocto project simplifyed the customizing of the embedded Linux system in maximum and saved many works, which let the developer can focus more on their works.

Key words: embedded; Linux; Yocto; open source

0　引言

1991 年芬兰大学生 Linus 发布了 0.1 版本的操作系统 Linux，到 1995 年，Linux 发布 1.2 版本，该版本开始支持 Alpha、i386、MIPS 和 SPARC 等架构，同时 MIPS 开始在低功耗设备中越来越流行[1]。1996 年第一家专注于嵌入式 Linux 的公司 Timesys 成立，随后基于嵌入式 Linux 的公司和组织如雨后春笋般出现[2]。

至今，基于 Linux 内核的操作系统广泛应用于消费电子、网络设备、工控控制、工业自动化、导航设备、宇宙飞船控制领域和医学设备领域[3]。

由于没有统一的标准，嵌入式 Linux 开发差异越来越大，导致大量重复工作，给嵌入式 Linux 开发者带来极大不便。在 Linux 基金会的有识之士的领导下成立了 Yocto 项目，Yocto 提供了工具、关键数据和方法来快速构建一个嵌入式 Linux 发行版操作系统，避免了大量的重复工作，这样嵌入式 Linux 开发者可以将更多的重心放在应用开发上，极大地提高了效率。

1　国内外研究现状

当前国内外开发一个嵌入式 Linux 操作系统的通用方式分为 3 步[4]：制作交叉编译工具链；移植 u-boot；配置并编译 Linux 内核[5]，制作根文件系统。这样的开发流程对于嵌入式 Linux 开发者来说非常繁琐，会耗费开发人员大量的精力在与应用开发上无关的方面，而且架构一换，交叉编译工具链还得重新进行配置和编译。即使有的公司或者社区已经做好了相应的工作，也仅仅限制于他们的芯片和板子，嵌入式 Linux 开发者需要查阅大量的文档才能将已有的工作移植到他们的工作中，因此对于开发者十分不利。

Yocto 项目正是针对这种情况而诞生的，它将构建嵌入式 Linux 操作系统中需要用到的软件源代码的下载、打补丁、配置、编译、打包和安装以 Python 或 Shell 脚本的方式描述出来，保存在以后缀为 .bb 的文件中，然后使用一个叫 bitbake 的工具来解析执行。这样的规则可以重利用，因此极大地提高了嵌入式 Linux 开发者的开发效率。Yocto 中提供了很多种类型的嵌入式 Linux 操作系统模版，涵盖了常见的操作系统类型，比如非常小的能启动设备的 core-image-minimal，而且使用者可以很容易地对该操作系统进行订制，仅仅通过修改构建目录下的 local.conf

文件就可以轻松完成各种需求的订制。

2 Yocto 项目介绍

Yocto 的核心是一个基于 OpenEmbeddedd 开放源代码的构建系统,它通过提供模版、工具和方法帮助开发者快速创建基于 Linux 内核的定制系统,支持 ARM、PPC、MIPS 和 x86 硬件体系结构。这个构建系统可以创建针对用户环境的嵌入式 Linux 操作系统发行版。Yocto 项目可以让用户订制不同大小的 Linux 镜像文件,从可以仅供启动设备的镜像到拥有各种各样功能的复杂的操作系统镜像文件。

除了可以订制嵌入式 Linux 操作系统发行版,还可以很容易地产生针对该嵌入式 Linux 操作系统的交叉编译工具链和 SDK,同时还有软件开发工具集,可以生成想要的格式,如:deb、rpm、ipk 等。

图 1 详细地描述了 Yocto 项目流程:从上游软件源将代码抓取下来,通过一个叫 bitbake 的构建系统,使用描述编译规则的文件,来生成内核的 image、根文件系统和 SDK。

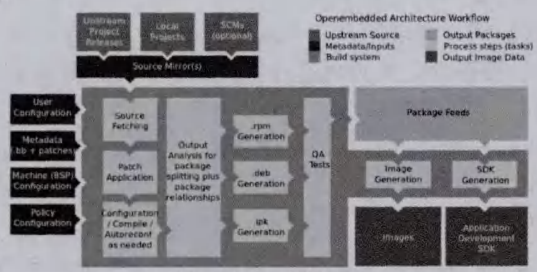

图 1 Yocto 项目流程

3 开发环境搭建

3.1 安装必要的软件包

Ubuntu14.04 下安装如下包:

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libsdl1.2-dev xterm
```

3.2 获取 Yocto 构建系统

```
$ git clone git://git.yoctoproject/poky
```

然后切换到当前最新的分支 jethro 上:

```
$ git checkout jethro
```

这样就得到了 Yocto 整个构建系统的所有工具、关键的配置文件和核心的元数据。

4 开始构建系统

接下来创建一个可以在 arm926ejs 的 CPU 上运行的嵌入式 Linux 操作系统发行版,还有一整套交叉编译工具链,并且让该操作系统运行在 QEMU 虚拟机上。

4.1 初始化构建系统的环境

poky 是一个包含构建系统、核心脚本文件和构建工具的一个集合。因此进入 poky 目录并运行如下脚本:

```
$ cd poky
$ source oe-init-build-env
```

该脚本会创建一个构建目录,默认放在 poky 里面,也可以自己指定构建目录,运行如下命令:

```
$ source oe-init-build-env you_path
```

构建目录包含构建期间生成的所有文件。

4.2 修改本地配置文件

在运行 oe-init-build-env 脚本后,会在构建目录下生成 conf 目录,conf 目录就是包含本次构建的所有重要配置文件。在 conf 目录下有一个 local.conf 的配置文件,该文件包含了要构建的系统的一些基本设置。

如果不改变任何配置文件就开始编译,默认的构建的目标机器是 qemux86,生成的可运行的镜像文件是基于 32 位的 x86 架构,可以运行在 QEMU 虚拟机上。对于本次编译来说,要生成基于 arm926ejs,可在 QEMU 虚拟机上运行的镜像文件,修改 local.conf 文件中的 MACHINE 变量:

```
MACHINE? = qemuarm
```

既然要做一个嵌入式 Linux 发行版,不是仅仅跑起来那么简单,作为一个成熟的嵌入式 Linux 操作系统,必须要有对应的包管理工具,默认使用 RPM,可以通过修改变量 PACKAGE_CLASSES 来改变:

```
PACKAGE_CLASSES = "package_deb"
```

当然还可以选择 ipk、tar 等方式。

4.3 bitbake 介绍

bitbake 是一个功能上与 make 类似的给嵌入式 Linux 操作系统生成可运行镜像文件和软件包的自动化构建工具,该工具受到了 Gentoo Linux 操作系统包管理工具 Portage 的启发,使用 Python 实现。

bitbake 操作的文件叫 bitbake recipe,以后缀.bb 结尾或者以.bbappend 结尾,该文件描述了一个软件包去哪下载、如何配置、如何编译、软件包的依赖、安装到哪里、如何删除等过程。

bitbake 的 recipe 文件可以解析多种软件包来源路径,包括:http、https、ftp、cvs、svn、git 和本地文件系统。在构建的过程中,recipe 文件用来跟踪软件依赖、执行本地和交叉编译,并且完成后将编译完的软件进行打包,可以在本地和目标设备上安装。

接下来使用 bitbake 生成一个包含根文件系统的嵌入式 Linux 镜像文件,bitbake 框架首先生成一个针对目标平台的交叉变异工具链。

4.4 构建嵌入式 Linux 操作系统

在 Yocto 中,针对不同的设备有相应的 recipe 文件,根据 recipe 文件的描述,可以生成大小不同的操作系统,有可以仅仅让设备跑起来实时响应的操作系统、带界面的操作系统等。接下来生成一个尺寸很小,刚好可以让设备跑起来的操作系统:

core-image-minimal

在4.1节搭建好的环境中运行：

$ source oe-init-build-env you_path

会生成you_path目录，这就是开发目录，在该目录下运行如下命令：

$ bitbake core-image-minimal

开始执行后，bitbake会根据当前的CPU型号生成一个交叉编译工具链，也就是生成一个可以在宿主机上运行的，能够编译出在目标板子上运行的程序的工具链，这样极大地简化了嵌入式开发者的工作，不需要自己手动修改代码来配置交叉编译工具链，仅仅通过配置就可以生成各种CPU型号的交叉编译工具链。使用该工具链编译各种在arm926ejs上可以运行的程序，包括Linux内核。

这是一个比较漫长的过程，因为针对每一个软件包都要下载、解压、配置、编译。通常bitbake会根据当前主机的CPU核心数量设置相应的线程来进行多线程编译。为了提高效率，同时防止在下载的过程中出现下载包失败的问题，通常可以提前下载好包，放在指定的目录下，然后在build目录中的local.conf文件中指定变量DL_DIR为上述指定的目录，这样bitbake构建系统就会去指定目录读取软件包，极大地提高了效率。编译操作系统如图2所示。

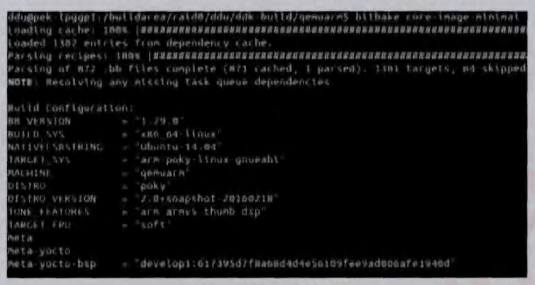

图2　编译操作系统

开始编译后，需要花费较长时间，具体时间长短取决于当前宿主机的CPU核的数量和当前内存大小。

4.5　运行生成嵌入式Linux操作系统

编译完成后，结果如图3所示。

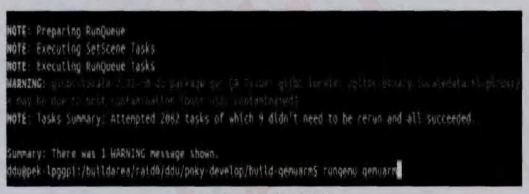

图3　编译完成

编译完成后，Linux内核的可执行镜像文件和根文件系统位于构建目录中的tmp/deploy/image/下。运行生成的嵌入式Linux操作系统，使用如下命令：

$ runqemu qemuarm slirp nographic

runqemu是一个启动QEMU虚拟机的脚本，qemuarm是运行的机器，slirp是一种不需要root权限的网络访问方式。runqemu脚本自动启动QEMU虚拟机，并且加载生成的内核和根文件系统，终端启动界面如图4所示。

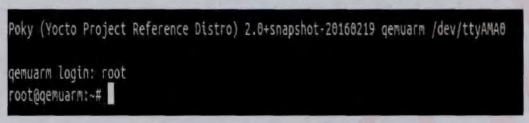

图4　登录Linux操作系统

使用root用户名登录，默认不需要用户密码。

5　结论

通过使用Yocto构建系统，仅需指定目标板子的类型和非常简单的设置，就可轻松地完成嵌入式Linux操作系统的构建，相比于传统的嵌入式Linux开发流程，极大地提高了效率，使开发人员能够避免底层复杂的配置和移植，将重心更多地放在自己的应用开发上。

Yocto项目可以订制操作系统的范围从非常小的传感器、智能手表到工控机，甚至复杂的服务器和大型机等，应用层面非常广泛，操作简单，仅使用Python和Shell脚本来描述操作系统的编译规则，极大地简化了开发工作。而且它可以订制基于多种架构的嵌入式Linux操作系统，由于从源代码到编译规则都是开源透明的，这对安全行来说十分重要，使开发人员对自己订制的操作系统百分百全面掌控。同时它减轻了嵌入式Linux开发人员的负担，使开发人员很容易地订制自己的操作系统，将更多的精力放在开发应用上。

Yocto是Linux基金会官方推荐的嵌入式Linux开发项目，未来，必将有更多的开发者和公司开始使用Yocto构建系统。

参考文献

[1] 刘庆. 嵌入式Linux技术的发展[J]. 开放系统世界，2003(9)：96-97.

[2] 苑庆国. 嵌入式Linux的实时风暴[J]. 开放系统世界，2004(10)：79-80.

[3] 陈闳中. Linux在嵌入式操作系统中的应用[J]. 同济大学学报(自然科学版)，2001，29(5)：564-566.

[4] 梁泉. 嵌入式Linux系统移植及开发技术研究[D]. 成都：电子科技大学，2003.

[5] 钱连举. 基于ARM的嵌入式Linux系统移植技术研究与应用[D]. 成都：电子科技大学，2006.

(收稿日期：2016-02-28)

作者简介：

杜登科(1990－)，男，硕士研究生，主要研究方向：嵌入式Linux系统开发。

胡爱兰(1973－)，女，硕士，高级工程师，主要研究方向：通信，信息处理及计算机应用。

李林峰(1989－)，男，硕士，助理工程师，主要研究方向：计算机软件开发。

嵌入式资源免费下载

总线协议：

1. 基于 PCIe 驱动程序的数据传输卡 DMA 传输
2. 基于 PCIe 总线协议的设备驱动开发
3. CANopen 协议介绍
4. 基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计
5. FPGA 实现 PCIe 总线 DMA 设计
6. PCI Express 协议实现与验证
7. VPX 总线技术及其实现
8. 基于 Xilinx FPGA 的 PCIE 接口实现
9. 基于 PCI 总线的 GPS 授时卡设计
10. 基于 CPCI 标准的 6U 信号处理平台的设计
11. USB30 电路保护
12. USB30 协议分析与框架设计
13. USB 30 中的 CRC 校验原理及实现
14. 基于 CPLD 的 UART 设计
15. IPMI 在 VPX 系统中的应用与设计
16. 基于 CPCI 总线的 PMC 载板设计
17. 基于 VPX 总线的工件台运动控制系统研究与开发
18. PCI Express 流控机制的研究与实现
19. UART16C554 的设计
20. 基于 VPX 的高性能计算机设计
21. 基于 CAN 总线技术的嵌入式网关设计
22. Visual C 串行通讯控件使用方法与技巧的研究
23. IEEE1588 精密时钟同步关键技术研究
24. GPS 信号发生器射频模块的一种实现方案
25. 基于 CPCI 接口的视频采集卡的设计
26. 基于 VPX 的 3U 信号处理平台的设计
27. 基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计
28. AT89C52 单片机与 ARINC429 航空总线接口设计
29. 基于 CPCI 总线多 DSP 系统的高速主机接口设计
30. 总线协议中的 CRC 及其在 SATA 通信技术中的应用
31. 基于 FPGA 的 SATA 硬盘加解密控制器设计
32. Modbus 协议在串口通讯中的研究及应用
33. 高可用的磁盘阵列 Cache 的设计和实现
34. RAID 阵列中高速 Cache 管理的优化

35. 一种新的基于 RAID 的 CACHE 技术研究与实现
36. 基于 PCIE-104 总线的高速数据接口设计
37. 基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计

VxWorks：

1. 基于 VxWorks 的多任务程序设计
2. 基于 VxWorks 的数据采集存储装置设计
3. Flash 文件系统分析及其在 VxWorks 中的实现
4. VxWorks 多任务编程中的异常研究
5. VxWorks 应用技巧两例
6. 一种基于 VxWorks 的飞行仿真实时管理系统
7. 在 VxWorks 系统中使用 TrueType 字库
8. 基于 FreeType 的 VxWorks 中文显示方案
9. 基于 Tilcon 的 VxWorks 简单动画开发
10. 基于 Tilcon 的某武器显控系统界面设计
11. 基于 Tilcon 的综合导航信息处理装置界面设计
12. VxWorks 的内存配置和管理
13. 基于 VxWorks 系统的 PCI 配置与应用
14. 基于 MPC8270 的 VxWorks BSP 的移植
15. Bootrom 功能改进经验谈
16. 基于 VxWorks 嵌入式系统的中文平台研究与实现
17. VxBus 的 A429 接口驱动
18. 基于 VxBus 和 MPC8569E 千兆网驱动开发和实现
19. 一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法
20. 基于 VxBus 的设备驱动开发
21. 基于 VxBus 的驱动程序架构分析
22. 基于 VxBus 的高速数据采集卡驱动程序开发

Linux：

1. Linux 程序设计第三版及源代码
2. NAND FLASH 文件系统的设计与实现
3. 多通道串行通信设备的 Linux 驱动程序实现
4. Zsh 开发指南-数组
5. 常用 GDB 命令中文速览
6. 嵌入式 C 进阶之道

7. Linux 串口编程实例
8. 基于 Yocto Project 的嵌入式应用设计
9. Android 应用的反编译
10. 基于 Android 行为的加密应用系统研究
11. 嵌入式 Linux 系统移植步步通
12. 嵌入式 CC++语言精华文章集锦
13. 基于 Linux 的高性能服务器端的设计与研究
14. S3C6410 移植 Android 内核
15. Android 开发指南中文版
16. 图解 Linux 操作系统架构设计与实现原理（第二版）
17. 如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核
18. Android 简单 mp3 播放器源码
19. 嵌入式 Linux 系统实时性的研究
20. Android 嵌入式系统架构及内核浅析
21. 基于嵌入式 Linux 操作系统内核实时性的改进方法研究
22. Linux TCP IP 协议详解
23. Linux 桌面环境下内存去重技术的研究与实现
24. 掌握 Android 7.0 新增特性 Quick Settings
25. Android 应用逆向分析方法研究
26. Android 操作系统的课程教学
27. Android 智能手机操作系统的研究
28. Android 英文朗读功能的实现

Windows CE：

1. Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计
2. Windows CE 的 CAN 总线驱动程序设计
3. 基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究
4. 基于 Windows CE.NET 平台的串行通信实现
5. 基于 Windows CE.NET 下的 GPRS 模块的研究与开发
6. win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码
7. Windows 下的 USB 设备驱动程序开发
8. WinCE 的大容量程控数据传输解决方案设计
9. WinCE6.0 安装开发详解
10. DOS 下仿 Windows 的自带计算器程序 C 源码
11. G726 局域网语音通话程序和源代码
12. WinCE 主板加载第三方驱动程序的方法
13. WinCE 下的注册表编辑程序和源代码
14. WinCE 串口通信源代码

15. WINCE 的 SD 卡程序[可实现读写的源码]
16. 基于 WinCE 的 BootLoader 研究
17. Windows CE 环境下无线网卡的自动安装
18. 基于 Windows CE 的可视电话的研究与实现

PowerPC：

1. Freescale MPC8536 开发板原理图
2. 基于 MPC8548E 的固件设计
3. 基于 MPC8548E 的嵌入式数据处理系统设计
4. 基于 PowerPC 嵌入式网络通信平台的实现
5. PowerPC 在车辆显控系统中的应用
6. 基于 PowerPC 的单板计算机的设计
7. 用 PowerPC860 实现 FPGA 配置
8. 基于 MPC8247 嵌入式电力交换系统的设计与实现
9. 基于设备树的 MPC8247 嵌入式 Linux 系统开发
10. 基于 MPC8313E 嵌入式系统 UBoot 的移植
11. 基于 PowerPC 处理器 SMP 系统的 UBoot 移植
12. 基于 PowerPC 双核处理器嵌入式系统 UBoot 移植

ARM：

1. 基于 DiskOnChip 2000 的驱动程序设计及应用
2. 基于 ARM 体系的 PC-104 总线设计
3. 基于 ARM 的嵌入式系统中断处理机制研究
4. 设计 ARM 的中断处理
5. 基于 ARM 的数据采集系统并行总线的驱动设计
6. S3C2410 下的 TFT LCD 驱动源码
7. STM32 SD 卡移植 FATFS 文件系统源码
8. STM32 ADC 多通道源码
9. ARM Linux 在 EP7312 上的移植
10. ARM 经典 300 问
11. 基于 S5PV210 的频谱监测设备嵌入式系统设计与实现
12. Uboot 中 start.S 源码的指令级的详尽解析

13. 基于 ARM9 的嵌入式 Zigbee 网关设计与实现
14. 基于 S3C6410 处理器的嵌入式 Linux 系统移植
15. CortexA8 平台的μC-OSⅡ及 LwIP 协议栈的移植与实现
16. 基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计
17. ARM S3C2440 Linux ADC 驱动
18. ARM S3C2440 Linux 触摸屏驱动

Hardware:

1. DSP 电源的典型设计
2. 高频脉冲电源设计
3. 电源的综合保护设计
4. 任意波形电源的设计
5. 高速 PCB 信号完整性分析及应用
6. DM642 高速图像采集系统的电磁干扰设计
7. 使用 COMExpress Nano 工控板实现 IP 调度设备
8. 基于 COM Express 架构的数据记录仪的设计与实现
9. 基于 COM Express 的信号系统逻辑运算单元设计
10. 基于 COM Express 的回波预处理模块设计
11. 基于 X86 平台的简单多任务内核的分析与实现
12. 基于 UEFI Shell 的 PreOS Application 的开发与研究
13. 基于 UEFI 固件的恶意代码防范技术研究
14. MIPS 架构计算机平台的支持固件研究

Programming:

1. 计算机软件基础数据结构 - 算法