文章编号:1005-1228(2013)06-0036-03

基于 S3C6410 处理器的嵌入式 Linux 系统移植

郝秉华

(内蒙古财经大学 计算机信息管理学院,内蒙古 呼和浩特 010070)

摘　要:文章详细阐述了如何在基于 S3C6410 微处理器的目标板上搭建嵌入式 Linux 系统的方法。首先介绍嵌入式 Linux 的移植方法,接着介绍交叉编译环境的搭建,最后详细讨论了将 BootLoader 、Linux 内核及根文件系统移植到 S3C6410 的过程。结果证明方法可行,重新搭建后的操作系统在目标板上运行稳定。

关键词:交叉编译;BootLoader;Linux 内核;根文件系统;移植

中图分类号: TP316.1　　**文献标识码**:A

The Transplantion of Embedded Linux System Based on S3C6410 Micro-processor

HAO Bing-hua

(Inner Mongolia University of Finance and Economics Department of Computer Information, Hohhot 010070,China)

Abstract: The article elaborates the method of how to build the embedded Linux system to the goal board based on the S3C6410 micro-processor in detail. Firstly,it introduces transplantion method for embedded Linux, and then introduces how to establish the cross compiling environment, finally, it discusses the transplantion process of the BootLoader, Linux kernel and Root filesystem to S3C6410 in detail. The result proves that the method is practical. The operating system builded runs stably on the goal board finally.

Key words: cross Compiling; BootLoader;Linux kernel;root filesystem;transplant

嵌入式系统是以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。[1]它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序等四个部分组成，用于实现对其他设备的控制、监视或管理等功能。目前,可以使用的嵌入式操作系统有很多,有 Linux、uClinux、WinCE、VxWorks 等。其中,Linux 操作系统是一种性能优良、源码公开且被广泛应用的免费操作系统。由于其体积小、可裁剪、运行速度快、安全性能高等优点，常被制作成嵌入式 Linux,广泛地应用于嵌入式系统中。嵌入式 Linux 是指对标准 Linux 经过小型化裁剪处理之后，能够固化在容量只有几 K 或者几 M 字节的存储器芯片或者单片机中，适合于特定嵌入式应用场合的专用 Linux 操作系统。[2]

本文叙述了嵌入式 Linux 操作系统开发环境的搭建以及在 AMR11 平台上的移植过程。

1　嵌入式 Linux 的移植方法

嵌入式 Linux 系统包括引导程序(BootLoader),内核(kernel)和根文件系统(Root filesystem)三个部分。在嵌入式系统的 Flash 存储设备上有相应的分区来存储它们,如图 1 所示。[3]

Boot Loader	参数	内核	根文件系统	应用程序

图 1　嵌入式 Linux 系统中的分区存储示意图

正常启动时,首先运行 BootLoader,引导程序将内核复制到内存中，并且在内存某个固定的地址设置好要传递给内核的参数,然后运行内核。内核启动之后,会挂接根文件系统,启动根文件系统中的应用程序。

在特定的硬件平台上搭建嵌入式 Linux 系统,一般需要以下几个步骤。

(1)建立交叉编译环境;

(2)编译、移植 BootLoader;

(3)配置和裁剪 Linux 操作系统内核,将其编译为镜像文件并下载到目标机;

(4)制作根文件系统,下载到目标机,可以在根文件系统中添加自己的应用程序。[1]

收稿日期:2013-08-25

作者简介:郝秉华(1980-),女,内蒙古呼和浩特人,讲师,主要研究方向:物联网研究。

2 搭建开发环境

嵌入式设备的资源有限，一般不能直接安装发行版的 Linux 系统，需要专门为特定的目标板定制 Linux 操作系统，这需要在主机上编辑系统程序，通过交叉开发模式进行编译，然后在目标板上运行程序。

本研究是用装有 VMware8.0+Fedoral14 虚拟机的 PC 机作为宿主机，目标机为 S3C6410 开发板。将宿主机与目标板通过串口线、网线、USB 线和 JTAG 线相连。从相关网站下载所需源码包 arm-linux-gccV4.3.1、u-boot-2011.09、linux-2.6.38.8.tar.bz2 和 busybox-1.19.2.tar.bz2。搭建好交叉编译的开发环境。

3 BootLoader 的移植

Boot Loader 就是在操作系统内核运行之前运行的一段小程序。通过这段小程序，我们可以初始化硬件设备、建立内存空间的映射图，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核准备好正确的环境。[3]

嵌入式系统中常见的 Boot Loader 有 Vivi、Blob、Redboot 和 U-Boot 等，其中 U-Boot 支持多种硬件构架，比如 ARM、X86 等；支持多种操作系统，比如 Linux、VxWorks 等；支持多达上百种以上的开发板，开放源代码，遵循 GPL 条款，易于移植与调试。鉴于上述特点，本文中选择 U-Boot 作为开发板的引导程序，但是 U-Boot 中没有对 S3C6410 开发板的支持，需要进行修改移植。下面对主要修改部分进行说明。

(1)首先在虚拟机 /home/uptech 目录下创建工作目录 u-boot，然后用 NFS 服务器将 u-boot-2011.09.tar.bz2 拷贝到虚拟机的 /home/uptech/u-boot 目录下，并解压。

#mkdir u-boot

#cp /mnt/hgfs/E/ u-boot-2011.09.tar.bz2 /home/uptech/u-boot

tar jxvf u-boot-2011.09.tar.bz2

解压后得到文件夹 u-boot-2011.09。

(2)该移植主要在 SMDK6400 开发板资源上添加和修改的，需要将 SMDK6400 下的相关文件复制到 u-boot-2011.09 文件夹下，并且把相应的头文件修改为支持 S3C6410 的头文件。

#cp include/configs/smdk6400.h include/configs/smdk6410.h

#cp arch/arm/include/asm/arch-s3c64xx/s3c6400.h arch/arm/include/asm/arch-s3c64xx /s3c6410.h

#cp board/samsung/smdk6400/ board/samsung/smdk6410/ -a

#mv smdk6400.c smdk6410.c

mv smdk6400_nand_spl.c smdk6410_nand_spl.c

修改 Makefile 文件，将 CONFIG_S3C6400 宏更改为 CONFIG_S3C6410 宏。

COBJS-$(CONFIG_S3C6410) += cpu_init.o speed.o

修改 common.h 头文件添加 CONFIG_S3C6410 宏处理。

defined(CONFIG_S3C6410) ||\

(3)对 SMDK6410 开发板进行参数配置，设置打开 CONFIG_CLK_667_133_66 时钟定义，修改 U-BOOT 引导内核参数等。

(4)修改完后，执行编译命令。

#make smdk6410_config

make

编译完成后，会在当前目录下生成 u-boot-nand.bin 二进制文件。通过烧写下载将映像文件烧到 Nand Flash 中，重启后可看到 U-Boot 信息。

4 Linux 内核移植

由于嵌入式系统资源有限，不能直接下载 Linux 系统使用，需要对 Linux 系统进行裁剪后移植。本文移植的是 Linux2.6 的内核，是稳定版本。

(1)首先在虚拟机 /home/uptech 目录下创建工作目录 kernel，然后用 NFS 服务器将 linux-2.6.38.8.tar.bz2 拷贝到虚拟机的 /home/uptech/kernel 目录下，并解压。

mkdir kernel

#cp /mnt/hgfs/E/linux-2.6.38.8.tar.bz2 /home/uptech/kernel

tar jxvf linux-2.6.38.8.tar.bz2

解压后得到文件夹 linux-2.6.38.8。

(2)进入 /home/uptech/kernel/ linux-2.6.38.8 目录，修改 Makefile 文件。

ARCH ?= arm

CROSS_COMPILE ?= arm-linux-

(3)仍然在 /home/uptech/kernel/ linux-2.6.38.8 目录下，拷贝参考配置文件，配置 SMDK6410 内核选项，并进行保存退出。

cp arch/arm/configs/s3c6400_defconfig .config

make menuconfig

(4)若不添加其它的外部设备的支持，就可以进行编译内核。

#make

最终会在内核源码目录 arch/arm/boot/ 目录下生成 zImage 内核镜像文件。由于我们前面使用的 bootloader 为 U-Boot，其默认引导 Linux 内核格式为 uImage，因此需要手动将 zImage 内核转换成 uImage 格式。经过 mkimage 工具转换，最终会在内核源码目录下生成 uImage 内核镜像文件。

uImage 内核镜像文件通过串口烧写到开发板，烧写过程需要几分钟的时间。

5 建立嵌入式 Linux 根文件系统

Linux 内核在系统启动期间进行最后的操作之一就是安装根文件系统，根文件系统一直是所有类 UNIX 系统不可或缺的组件。busybox 是很小的一个应用程序，提供了完整的制作根文件系统的工具集，本文使用 busybox 制作根文件系统。在路径 /home/uptech 下创建 rootfs 文件夹，在此文件夹下存放根文件系统。

(1)在 /home/uptech/rootfs 解压源码包。

tar xjvf busybox-1.19.2.tar.bz2

(2)进入 busybox-1.19.2 目录，进行配置。

make menuconfig

根据具体的需要添加或去掉某些项，其主要是指定交叉编译器路径和 busybox 要运行的平台类型，进行完相关配置后保存退出。

(3)编译。

make

make install

执行完后，将在 rootfs 目录下生成根文件系统用到的 /bin、/sbin、/linuxrc、/usr、/usr/bin、/usr/sbin 目录，其中包含的是可以在目标平台上运行的命令。再建立其它目录，如：dev、etc、lib、var、root、mnt、proc、tmp、sys，并给相关目录填充必要的内容。dev 目录下包含设备文件及其他特殊文件；etc 目录下包含了系统配置文件；lib 目录下是必要的链接库，像 C 链接库、内核模块等；var 目录下包含了监控程序和工具程序存放的可变数据等。

根文件系统准备好内容之后，就需要给根文件系统选型了，Linux 支持多种文件系统，常见的文件系统有 tmpfs、cramfs、ext2/ext3、ffs2 和基于 Nand Flash 的 yaffs 等。本文选用只读压缩的 cramfs 文件系统，制作 cramfs 只读根文件系统映像文件并烧写启动。

6 结束语

本研究所做的工作是对 U-Boot 做了一些修改，使其在 S3C6410 微处理器上顺利地启动，对实时性强、可靠性强的 Linux2.6.38 内核进行了裁剪，编译生成的内核移植到目标板上能成功运行，制作的根文件系统能正常加载进内核，为后续的开发提供了一个良好系统软件，之后就可以根据需要加载驱动程序，编写应用程序，使系统在后续的物联网研究应用中具有实用价值。

参考文献：

[1] 王孝良,刘全利,赖晓晨,等.基于 ARM 平台的嵌入式核心编程[M].北京:清华大学出版社,2011.

[2] 朱华生,吕莉,熊志文,等.嵌入式系统原理与应用——基于 ARM 微处理器和 Linux 操作系统[M].北京:清华大学出版社,2012.

[3] 吴晓云,冯兴乐.基于 S3C2440A 的嵌入式 Linux 系统的搭建[J].微计算机信息,2010,26(6-2):108-110.

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

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

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

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

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

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

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