

文章编号: 1005—0523(2006)02—0111—03

ARM Linux 在 SkyEye 上的移植

宋 凯, 严丽平, 甘 岚

(华东交通大学 信息工程学院, 江西 南昌 330013)

摘要:介绍了一款优秀的嵌入式仿真软件—SkyEye, 描述了其硬件逻辑结构及执行流程, 并详细讨论了嵌入式 OS—ARM Linux 在其上的移植过程.

关 键 词: ARM Linux; SkyEye; 移植
中图分类号: TP316.2 **文献标识码:** A

0 引言

随着市场需求和嵌入式仿真软件 SkyEye 的日益成熟, 众多的嵌入式软件工程师开始选择 SkyEye 作为软件设计、开发与调试的平台. SkyEye 平台是硬件开发板的变异与延伸, 它采用软件技术模拟真实的硬件环境. 其各个硬件仿真模块已由开发者完备地验证过, 因此基于 SkyEye 平台的软件设计可以忽略大量的硬件细节, 获得更高的设计效率, 软件产品的设计周期也得以缩短.

同时随着应用程序规模的不断扩大, 在 SkyEye 平台上运行操作系统的需求也越来越迫切. 操作系统可以大大减轻程序员在开发图形界面、多任务调度、硬件资源管理等复杂应用中的负担. 而 ARM Linux 正是一款优秀的嵌入式实时操作系统^[1], 所以 ARM Linux 在 SkyEye 平台的移植有一定的现实意义.

1 ARM Linux 操作系统简介

ARM Linux 源自标准的 Linux 内核, 主要针对 ARM 核的处理器进行了相关移植. 由于 ARM 核的

处理器在嵌入式系统领域应用十分广泛, 因而 ARM Linux 也成为诸多嵌入式系统的操作系统平台. 为了适应嵌入式应用的需求, 它采用了 romfs 文件系统, 可执行文件格式采用的是 FLAT 格式, 而不是标准 Linux 中采用的 elf 格式. 相对于另一个嵌入式的操作系统 uClinux^[2], 它启动了 MMU, 应用程序不能直接读写物理地址(包括 CPU 寄存器、MEMORY、外设芯片寄存器等), 而必须借助驱动的形式, 切入内核用 ioremap 来实现对这些物理地址的访问^[3].

ARM Linux 的源代码是完全公开的, 并且绝大部分源代码都是采用 C 语言, 只有那些直接与 CPU 相关的代码比如开关 CPU 中断等操作才使用汇编语言, 因此 ARM Linux 具有很好的移植性^[4].

2 SkyEye 仿真平台介绍

SkyEye 是一个开源软件项目, 中文名字是“天目”. SkyEye 的目标是在通用的 Linux 和 Windows 平台实现一个纯软件集成开发环境, 模拟常见的嵌入式计算机系统. SkyEye 是一个指令级模拟器, 可以模拟多种嵌入式开发板, 可支持多种 CPU 指令集, 在 SkyEye 上运行的操作系统意识不到它是在一个虚拟的环境中运行, 而且开发人员可以通过 SkyEye

收稿日期: 2006—01—08

作者简介: 宋凯(1980—), 男, 河南泌阳人, 华东交通大学在读研究生, 研究方向: 嵌入式系统.

调试操作系统和系统软件。

SkyEye 从总体上分为四个层次:

1) 用户接口模块:完成处理用户的输入命令,并把相关调试数据输出给用户的任务。

2) 符号处理模块:主要处理执行文件的头信息,解释执行文件中内嵌的调试信息等。

3) 目标控制模块:主要完成执行控制,程序栈结构分析,对具体目标硬件的控制。

4) 目标模拟模块:是 SkyEye 的核心,它模仿计算机系统的主要硬件的执行,对执行文件的机器指令解释,并模拟执行每一条机器指令,产生相应的硬件响应等。

SkyEye 起源于 GDB/ARMLulator,并进行了改进和扩展。目前模拟的硬件包括 CPU、内存、I/O 寄存器、时钟、UART、网络芯片、MMU、CACHE 等^[5]。其硬件模拟逻辑结构如图 1 所示。

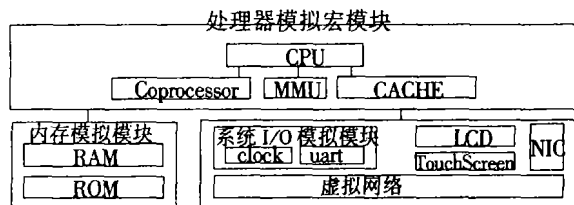


图 1 SkyEye 硬件模拟逻辑结构

3 ARM Linux 操作系统的移植

在移植工作开始之前,应该先做好准备工作,即准备好移植需要的软件包。移植主要需要三个软件包:ARM Linux 源码包、ARM 交叉编译所需的编译工具包以及 busybox 工具包。上述三种工具包都有众多版本,本文中选择的分别是 linux-2.4.18-mm7.tar.bz2、cross-2.95.3.tar.bz2、busybox-1.00.pre10.tar.gz。移植工作主要分 4 步完成:建立交叉编译环境;编译内核;生成并配置根文件系统;配置 SkyEye 系统信息。下面详细介绍整个过程。

3.1 建立交叉编译环境

由于已经准备好了所需软件包,所以建立交叉编译环境的过程实际就是对 cross-2.95.3.tar.bz2 解包的过程。在本文中,所有的软件源码包都放在 tools 目录下,解包所用指令序列在此不做赘述,其解包路径为/usr/local/arm__cross。在上述路径下完成解包后,需使用 export 指令设置交叉编译工具路径。至此,交叉编译环境已经建立起来。

3.2 交叉编译 ARM Linux 内核

在交叉编译内核之前,对编译选项的配置是很

重要的。执行“make menuconfig”指令,进入 Syetem Type 选项。本文中选择对 CLEP7312 系统板的支持,然后配置 File System,对于 File System 应确保如下支持:

1 /proc file system support

2 ROM file system support

3 Second extended fs support

最后配置 Block device,确保其有如下支持:

4 RAM disk support

5 Initital RAM disk (initrd) support

以上配置是内核编译的必备工作,保存配置后要修改 Makefile 文件,即把交叉编译的路径指定为本文中安装的特定目录,修改语句为:

```
CROSS __Complie=/usr/local/arm__cross/2.95.3/bin/arm-linux-
```

交叉编译工作由一下指令完成:

```
$ make dep //设置依赖关系
```

```
$ make //编译内核
```

交叉编译时间相对较长,最终会生成一个文件 vmlinux,这就是编译成功后的 ARM Linux 内核文件。本文中所有的测试工作都是在/skyeye-test 目录下完成,所以还应该复制 vmlinux 文件到该目录下,以备测试。

3.3 制作简单的文件系统并配置根文件系统

1) 安装 busybox 工具包

制作系统文件的工具包是 busybox^[6],所以在制作之前应该先安装 busybox 工具。本文中使用的软件包是 busybox-1.00.pre10.tar.gz。首先将其解压,在解压目录下修改 Makefile 文件,将 DOSTATIC 的值改为 ture,另外还要修改 init.c(即启动文件),修改格式如下:

```
#ifndef INIT __SCRIPT
#define INIT __SCRIPT "/etc/ inittab"
#endif
```

运行如下指令完成安装工作:

```
$ make TARGET __ARCH=arm CROSS=arm-linux- PREFIX=/tmp/target all install
```

2) 创建文件系统的映象文件

①在/mnt 下创建 img 目录,然后进入根目录执行如下指令创建一个空的映象文件:

```
$ dd if=/dev/zero of=initrd bs=1024k count=1
$ mke2fs -F -m0 initrd
```

②使用如下指令加载创建的文件系统,并复制文件系统中必要的文件:

```
$mount -t ext2 -o loop initrd /mnt/img
$cp -Rd /tmp/target/* /mnt/img
```

3)配置根文件系统

经过上述步骤已经产生了文件系统所需要的一些文件,但是需要设置根路径,即在根路径下添加必须的目录、文件并且正确设置属性.

①创建目录

在/mnt/img 目录下创建根文件系统的一般目

录/bin /etc /proc /tmp /var /dev /mnt,并使用 chmod 命令改变其属性.

②创建设备文件

建立设备文件,一个简单的方法是复制主机/dev/目录下一些必要的设备文件到/mnt/img/dev/目录下.基本的设备文件有:/dev/console /dev/fd0 /dev/null /dev/am0 /dev/tty /dev/tty0.在复制的过程中,要注意使用 dp 参数以保证文件属性不变.

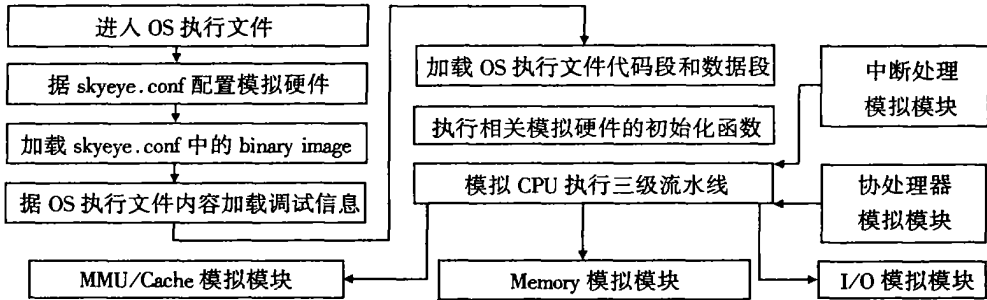


图 2 SkyEye 逻辑执行流程

③创建相关启动配置文件

在/mnt/img/etc/目录下创建启动配置文件 init-tab,编辑内容如下:

```
::sysinit:/etc/rc
```

在/target/etc/目录下创建启动脚本 rc,编辑内容如下:

```
#!/bin/sh
hostname ARMTTest
mount -t proc proc /proc
mkdir /var/tmp /var/log /var/lock
cat /etc/motd
```

在/mnt/img/etc/目录下建立开发板上的文件系统配置文件fstab,编辑内容如下:

```
none /proc proc defaults
```

上述工作完成后,即可使用 umount 命令结束操作,并将生成的文件系统 initrd 复制到测试目录.

3.4 配置 SkyEye 系统信息

在装载之前,首先了解 SkyEye 逻辑执行过程,如图 2 所示.从图中可知是 skyeye.conf 文件仿真环境启动的必备配置文件,应该首先新建并编辑它.根据编译内核时选择的系统板,其内容应作如下配置:

```
cpu: am720t
mach: ep7312
mem __ bank: map = I, type = RW, addr =
0x80000000, size=0x00010000
mem __ bank: map = M, type = RW, addr =
```

```
0xc0000000, size=0x00028000
mem __ bank: map = M, type = RW, addr =
0xc0028000, size = 0x001d8000, file = ./vmlinux, boot
=yes
mem __ bank: map = M, type = RW, addr =
0xc0200000, size=0x00200000, file=./initrd
mem __ bank: map = M, type = RW, addr =
0xc0400000, size=0x00c00000
```

4 装载 ARM Linux 并进行测试

在上述配置完成以后可以执行如下指令完成装载:

```
$skyeye //进入 skyeye
(SkyEye)target sim //读取配置文件,并装入内核
(SkyEye)load //加载模块
(SkyEye)run //在 skyeye 中运行
```

以上的指令序列将会启动 SkyEye 仿真软件,并装入内核并运行之.如果运行结果出现图 3 所示画面,则说明移植已初步成功.

为了进一步验证移植的稳定性,本文基于 ARM Linux 内核配置了 Web 服务器,具体过程如下:

1) 下载 boa 源码包,解压并编译之,将生成可执行文件 boa.在编译之前需要在/boa/src/configure 文件中加入 CC 和 CPP 的说明,同时修改 Makefile 文件中交叉编译器选项.

2) 建立目录. 由于默认的根文件系统 ramdisk 是只读的, 不能用 mkdir 等命令来新建目录, 应在编译内核前先建好要用到的目录.

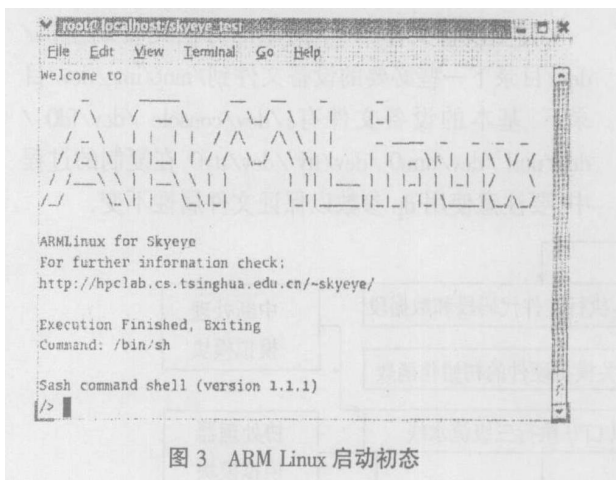


图 3 ARM Linux 启动初态

3) 修改配置文件 boa.conf. 进入 /boa/src 目录, 修改 defines.h 文件, 将根目录定义语句改为: #define ERVER __ROOT "/home/httpd".

4) 在 ramdisk 中的 /bin/ 下加入 boa 可执行文件. 并把修改后的 boa.conf 和 mime.types 拷贝到 Web 服务器根目录 /home/httpd 下. 并将一些静态页面放在由 boa.conf 指定的目录下.

5) 配置完成后, 重新编译内核. 装载内核到 SkyEye, 启动 Boa Web Server.

通过浏览器可以正确访问刚刚配置的 Web

Server, 网页也可以正常显示. 进一步验证了移植的稳定性.

5 结束语

目前, 笔者移植的 ARM Linux 已经能稳定地运行在 SkyEye 仿真平台上, 这使得基于 SkyEye 平台的嵌入式 Gui 的移植以及应用程序的开发有了 OS 基础, 为嵌入式软件设计节约了大量的开发时间, 大大提高了开发效率, 是后续嵌入式软件开发的有力支持. 当然 SkyEye 只是一个仿真平台, 涉及到硬件环节的开发工作, 还需要嵌入式开发板的支持.

参考文献:

- [1] 郑浩, 等. ARM Linux 中断处理实时性能分析. 计算机工程与应用[J]. 2005, 41(19): 101—104.
- [2] 沈沙, 等. uclinux 操作系统在嵌入式 SOC 平台上的移植. 计算机工程与应用[J]. 2004, 40(26): 104—108.
- [3] 占文静, 等. uclinux 在 S698 处理器上的移植研究与实现. 计算机应用[J]. 2004, 25(5): 55—57.
- [4] 张永强, 等. 基于 ARM 的 HMS30C7202 平台的嵌入式 Linux 移植. 微计算机信息[J]. 2005, 21(4): 125—126.
- [5] 陈渝, 等. 源码开放的嵌入式系统软件分析与实践[M]. 北京: 北京航空航天大学出版社, 2004.
- [6] Karim Yaghmour. Building Embedded Linux System. O'Reilly. 2004.

Analysis of Porting ARM Linux to SkyEye

SONG Kai, YAN Liping, GAN Lan

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: This paper introduces an excellent embedded emulational software-SkyEye and describes its hardware logic structure and operating process. Then it discusses on how to port ARM Linux to SkyEye in detail.

Key words: ARM Linux; SkyEye; port