

# 嵌入式操作系统 $\mu$ Clinux 移植研究

丁苍峰

(延安大学 计算中心, 陕西 延安 716000)

**摘 要:**为开发一个稳定性好、适应性强和安全性高的适合多嵌入式操作系统平台的嵌入式 Web 服务器,搭建  $\mu$ Clinux 移植需要的开发配置环境和设计移植的系统流程,讨论了移植  $\mu$ Clinux 需解决的如交叉编译环境的建立、 $\mu$ Clinux 的内核加载方式、改写编译 blob.bin 和烧 blob 等一些关键性技术问题,对  $\mu$ Clinux 移植做了恰当的裁减和分析工作以适合在设计平台上的移植。并对移植  $\mu$ Clinux 需要的烧 bootloader 到 Flash、烧  $\mu$ Clinux 内核文件到 Flash、烧 romf 只读文件系统到 Flash 的分析研究及根文件系统的烧到 Flash 进行了详细过程描述。

**关键词:**移植;  $\mu$ Clinux; 烧; 嵌入式; Web 服务器

**中图分类号:**TP393.03

**文献标识码:**A

**文章编号:**1673-629X(2011)10-0077-04

## Transplant Research of Embedded Operating System $\mu$ Clinux

DING Cang-feng

(Compute Center of Yan'an University, Yan'an 716000, China)

**Abstract:**To develop a stable, strongly adaptive and highly security embedded Web server for adapting well to multi-platform embedded operating systems, build development configuration environment and system process of transplant  $\mu$ Clinux, it discusses some key technical problems which are cross-compilation environment establishment,  $\mu$ Clinux kernel loading method, rewriting and compiling of blob.bin and burning blob, does appropriate cuts and analysis for transplant  $\mu$ Clinux in order to suit for the transplant in design of the platform, and describes the needs of transplant  $\mu$ Clinux which are burning bootloader to Flash, burning kernel file to Flash, burning read-only romfs file system to Flash and root file system burning to Flash in detail.

**Key words:**transplant;  $\mu$ Clinux; burning; embedded; Web server

## 0 引言

嵌入式 Internet 技术带动了设备和设备之间、设备和人之间的通信方式的变革,而研究实现嵌入式 Internet 的关键技术之一是嵌入式 Web 服务器的实现技术,对于嵌入式 Web 服务器的研究和应用,其意义是重大的。它为人们管理、控制和监测各种各样的设备提供了一个很好的途径。把 Web 服务器应用到嵌入式设备已经是可以实现的技术,但一个稳定性好的、适应性强的、安全性好的、可以适合多嵌入式操作系统的和有自己知识产权的嵌入式 Web 服务器软件还是一个市场的需求<sup>[1]</sup>。要实现这样一个嵌入式 Web 服务器,其首先必要的任务就是对熟悉的多种嵌入式操作系统如  $\mu$ Clinux 进行移植研究,开发出一个只要稍加修改就可以在不同平台上移植的嵌入式 Web 服务器。

今天的嵌入式系统中,嵌入式操作系统已经成为

嵌入式系统中不可缺少的系统软件。大多数的操作系统都有一个硬件抽象层,用来向上层隐藏具体硬件的细节,操作系统的大部分模块在一个统一的抽象硬件模型中工作。对这个抽象的硬件模型的研究是对一个完整的操作系统探索的一部分。嵌入式操作系统不但要尽可能维护对上层的硬件抽象,还要充分利用所使用的嵌入式微处理器的特性,尤其是充分发挥电源管理模块的性能,以提高整体性能,这种对新的矛盾的平衡点的摸索,不可避免需要深入到具体的硬件细节中去,正是嵌入式的特点决定了将嵌入式操作系统和具体的硬件结构放在一起研究的必要性。

论文的初衷是希望研究一个适合多平台的嵌入式 Web 服务器,因此需要对多种平台的调研。在资料调研过程中,逐步选定几种典型嵌入式操作系统,如  $\mu$ Clinux<sup>[2]</sup>、嵌入式 Linux<sup>[3]</sup> 和 Windows CE<sup>[4]</sup> 操作系统。 $\mu$ Clinux 是近年来伴随着 ARM 系列微处理器的流行,在标准的 Linux 的基础上,由诸多 Linux 爱好者开发的针对无内存管理单元微处理器的开源嵌入式操作系统。它保留了 Linux 大部分的系统调用,大部分的 Linux 应用层软件包不需要任何的改动或只需要经

收稿日期:2011-03-12;修回日期:2011-06-20

基金项目:陕西省教育厅教学改革项目(09BY37)

作者简介:丁苍峰(1978-),男,河南唐河人,CCF 会员,硕士,讲师,研究方向为嵌入式和工作流技术。

过少许的改写,经过重新编译就可以在  $\mu\text{Clinux}$  上运行<sup>[5]</sup>。选择  $\mu\text{Clinux}$  作为分析研究的对象,可以为其他功能更为完善的 Linux 变种的移植<sup>[6]</sup>打下坚实的基础。课题  $\mu\text{Clinux}$ 、Windows CE 及 Linux 嵌入式操作系统是在 NETARM3000<sup>[7]</sup>硬件平台上移植完成的,在此基础上实现了多平台的嵌入式 Web 服务器。

嵌入式 Web 服务器实现如图 1 所示,首先在 UP-NETARM3000 硬件平台上移植  $\mu\text{Clinux}$  操作系统,在  $\mu\text{Clinux}$  操作系统上实现完成嵌入式 Web 服务器<sup>[8]</sup>后,对编码进行改进,可以很容易地改到 Linux 内核的操作系统上,然后调试程序满足 Windows 操作系统,这时就可以再一次改进代码适合 Windows CE 嵌入式操作系统。

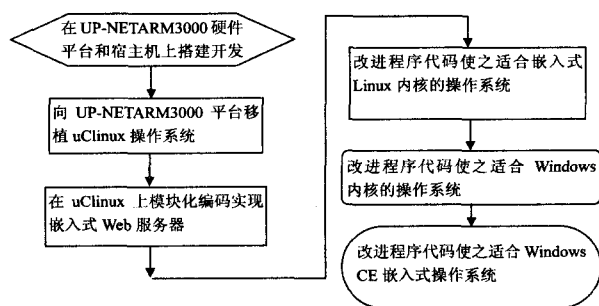


图 1 系统开发流程图

## 1 移植 $\mu\text{Clinux}$ 需解决的关键技术问题

### 1.1 建立 $\mu\text{Clinux}$ 交叉编译环境

由于嵌入式  $\mu\text{Clinux}$  系统开发资源有限<sup>[9,10]</sup>,不可能在开发板上同时进行开发和调试工作,所以要采用建立  $\mu\text{Clinux}$  交叉编译环境,交叉编译环境的目标采用“目标机(开发板)和宿主主机”开发方式,这两种机子一般采用串口连接方式,也可通过网络端口的方式连接,同时  $\mu\text{Clinux}$  必须支持目标机的处理器体系结构。针对种类众多的处理器,GNU 提供了众多的开发包。可以通过利用网络资源下载所需的开发包,从网上下载的资源,部分是源代码包,这些源代码包必须在 gcc 编译后才可使用,部分是直接可以安装的应用程序包。把上述的资源安装到机子后,开发所用的 Linux 系统中就有了基于处理器体系结构的编译、链接、调试器和使用工具,在以后的开发工作中,例如  $\mu\text{Clinux}$  内核的加载、文件系统的制作都是要在  $\mu\text{Clinux}$  交叉编译环境下进行。

### 1.2 加载 $\mu\text{Clinux}$ 内核方式

有两种方式可以运行  $\mu\text{Clinux}$  内核:一种是直接在 Flash 上运行,另一种在内存中加载运行。针对这两种方式而言,在 Flash 上直接运行的优点就是需要很少的内存。内存中加载运行方式则把压缩的内核文件放在 Flash 上,系统启动过程中会读取压缩在 Flash

上的文件并把该文件解压在 RAM 中,从而启动执行。相对前一种方式,这种做法要复杂些,由于内存 RAM 的存取速度比闪存 Flash 高的原因而使加载到内存中运行的速度可能更快些。标准配置下的 Linux 系统就采用这种方式启动,这种启动方式需要编写一个软件 Bootloader<sup>[11]</sup>。

理论上, $\mu\text{Clinux}$  引导时并非一定需要一个独立于内核的 Bootloader。然而,将 Bootloader 与 Kernel 分开设计能够使软件架构更加清晰,也有助于灵活地支持多种引导方式,实现一些有用的辅助功能。Bootloader 的主要任务包含硬件初始化和系统引导,加载 Kernel Image(如果需要)、设置需要传递给 Kernel 的启动参数(如果需要)和调用  $\mu\text{Clinux}$  Kernel 等。

### 1.3 改写编译 blob. bin 和烧 blob 的问题

(1)在 blob 代码中对 Flash 空间的分区问题。NETARM3000 硬件平台采用 2M NOR Flash 作为 BIOS,而要求将 bootloader、 $\mu\text{Clinux}$  内核和只读的文件系统固化在 NOR Flash 中。这样要对 Flash 的地址空间进行分区,分别存放 bootloader、 $\mu\text{Clinux}$  内核和文件系统。分区的方式一般有两种:第一种是根据 bootloader、 $\mu\text{Clinux}$  kernel 和 file 系统计划预订的存储空间,允许它们拥有自己固定的首地址和分区。第二种就是按照 bootloader、 $\mu\text{Clinux}$  kernel 和 file 系统实际情况分配空间,这种分区方法就是它们分配的空间是连续的存储单元,分配的结果是 bootloader、 $\mu\text{Clinux}$  kernel 和 file 系统没有固定的分区和首地址。如果在空间足够的情况下,通常采用第一种方式,虽然可能会浪费一部分 Flash 空间,但在改写 bootloader 程序时,方便内核的加载和文件系统的挂载。

(2)XMODEM 协议操作过程中的间隔时间问题。从串口下载  $\mu\text{Clinux}$  内核和 ROMFS 的过程中用到了 XMODEM 协议,串口通信有个 TIMEOUT 值,为了要求这个值的合理性,故修改了 linux2.4. x/include/asm/arch/uncompress. c 中相应的函数。

(3)烧 Flash 的程序应注意的问题。烧 Flash 的程序具体过程必须要参看 SST39L(V) F160 中文数据手册。由于对 Flash 的写入和擦除操作都必须严格地按照规定步骤的顺序地进行,要对端口操作时就必须首先关闭 cache,并且也要管理编译器,目的就是为了防止编译器对程序语句所在的一切优化,否则造成了实际的操作顺序并不是事先预订好的,从而导致了操作无用。

### 1.4 $\mu\text{Clinux}$ 应用程序在嵌入式系统中运行的问题

在嵌入式应用中,一般需要在操作系统运行起来后立刻运行用户的特定用户程序,以往开发的嵌入式应用程序都放在了只读的文件系统中,这样产生的问

题是,如果程序有改动则要求重烧 Flash,所以本课题将应用程序放到了可读写的文件系统中,但要求对操作系统内核初始的配置进行改动。首先要求分析操作系统启动后最后的初始化过程,然后本课题编写了一个 myhttp. sh 文件,解决了这个问题。

## 2 移植 $\mu$ CLinux

首先要求准备好 bootloader<sup>[12]</sup>、 $\mu$ CLinux 内核和只读的 romfs 文件系统,它们分别为 blob. bin、zImage 和 romfs. bin,将它们烧到 Flash 中后,整个  $\mu$ CLinux 操作系统在 NETARM3000 硬件平台上移植完毕。通过对软件平台原理的分析,完全可以到开源代码网站下载这些源程序代码,然后裁减适合自己的硬件平台,利用开发工具编译完成 Blob、 $\mu$ CLinux 内核和 romfs 文件系统。或者可以到官方网站下载已经编译好的比较通用的软件平台,但要求必须完全理解软件平台工作原理和已经编译好的软件平台的配置要求,否则可能无法满足系统需求。

### 2.1 烧 bootloader 到 Flash

首先要求搭建好宿主机的开发环境。采用硬件设备 JTAG 仿真器或较昂贵的 UP-ICE100 仿真器一面连接宿主机一面连接 NETARM3000 硬件平台。要求宿主机安装好 JTAG 或 UP-ICE100 的驱动程序和应用程序。要求完全掌握 JTAG 或 UP-ICE100 的烧芯片应用程序的理论和各种功能,才不至于对所烧的芯片进行误操作。连接好 UP-ICE100 仿真器,打开电源,等待红色。

如图 2 主界面所示,单击初始化配置按钮(或者执

行菜单中的相应命令),弹出仿真器配置对话框。选择正确的串口,单击读取 ICE 状态按钮,可以读取 UP-ICE 的配置信息,包括:IP 地址、子网掩码、网关、启动模式和配置存储器的信息。其中,启动模式包括:

(1)复位 & 运行。仿真调试或者对目标板 Flash 编程的时候,自动复位目标板。

(2)继续运行。仿真调试或者对目标板 Flash 编程的时候,不对目标板作任何操作。

(3)复位 & 停止在 xxx ms。仿真调试或者对目标板 Flash 编程的时候,自动复位目标板,并等待 xxx 毫秒以后,停止目标板运行。

根据不同的需要,可以使用不同的启动模式。通常,目标板已经有了正确的初始化配置程序以后,在调试程序的时候,通常使用模式复位 & 停止在模式,这种模式可以保证在调试程序之前,自动复位并运行一段目标板的初始化程序,以正确地配置目标板的工作模式。在启动模式中,还包括对于目标板处理器的寄存器的配置。开启允许初始化存储器选项,可以给设定的地址添加数据,在目标板启动的时候,由 Up-ICE100 自动配置到目标板上,以配置目标板处理器的工作模式。在对目标板的 Flash 编程的时候也是必不可少的。

完全理解上面的工作原理后,最后对 bootloader 进行 Flash 编程。要求首先读入 blob. bin 文件。然后通过选择 Flash 按钮可以选择 Flash 型号。设置好 Flash 和目标板的相关信息(比如,Flash 编程的起始地址,ARM 外部总线的宽度等信息)。使用整片擦除或者扇区擦除按钮,实现对 Flash 的擦除。扇区擦除的时候

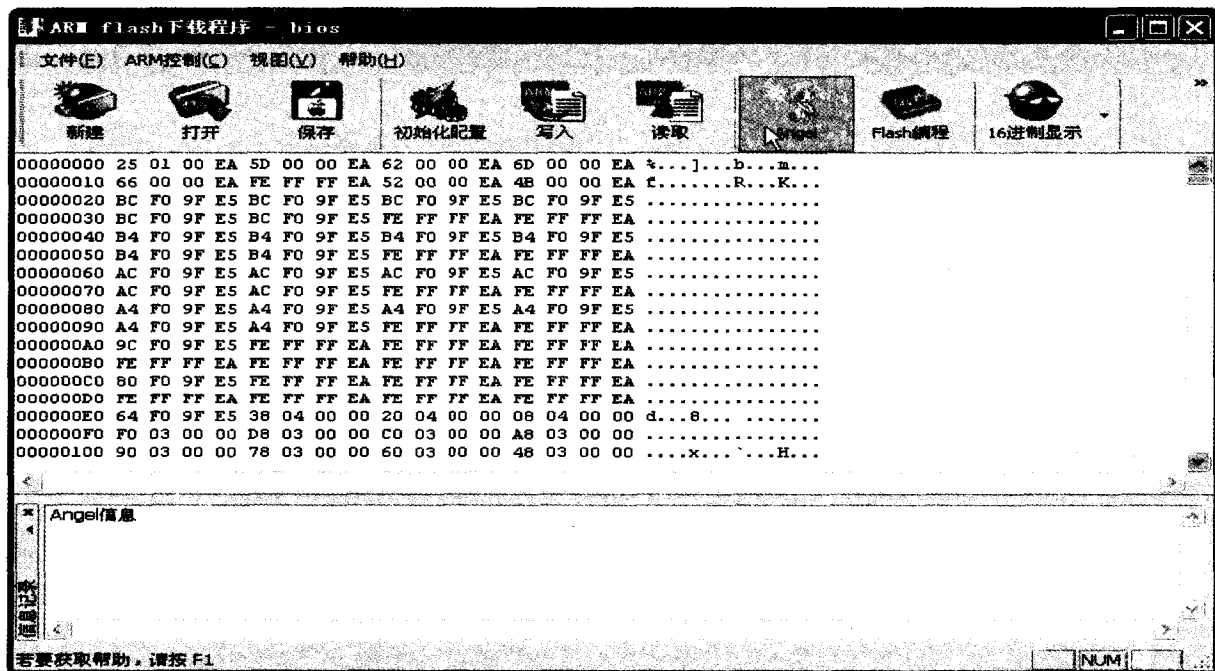


图 2 烧 bootloader 到 Flash 程序的主界面

软件可以根据用户选择的 Flash 扇区的大小、所要编程的文件大小以及起始地址等信息,自动计算出所需要擦除的扇区并给与提示。单击编程按钮,即可开始对 Flash 编程,将 blob. bin 编程到 Flash 后,可以将整个系统完全重启后从 0x00 地址读取内容,然后与 blob. bin 内容比较,可以检查写入的 Blob 是否是正确的。

## 2.2 烧 $\mu$ Clinux 内核文件到 Flash

在 Linux 平台下进行  $\mu$ Clinux 内核和根文件系统的烧到 Flash 过程有以下几步:

(1)启动 Linux 下的超级终端(minicom),对其进行正确配置。在 Linux 平台的 X window 界面下建立一个终端(右键单击屏幕——>新建终端),在终端的命令行提示符后键入 minicom,回车,你就会看到 minicom 的启动画面。若没有启动 X window 则在命令行提示符后直接键入 minicom。minicom 启动后,先按 Ctrl+A 键,再按 Z 键(注意不是连续按,Ctrl+A 松开后才按 Z),再按"O",进入配置界面。按上下键选择 Serial port setup,进入端口设置界面,这里有几个重要选项改为如下值(在 Change which setting 后按哪个字母就进入哪项的配置,如按 A 进行端口号配置):

A——Serial Device :/dev/ttyS0 (端口号使用串口 1)

E——BPS/par/bits:/115200 8N1 (波特率)

F,E 硬件流,软件流都改为 NO。若要使用 PC 机的串口 2 来接板子的串口 1 做监控,改为:/dev/ttyS1 即可。

(2)选好后按 ESC 键退出,选择 Save setup as df1 保存退出,以后只要启动 minicom 就是该配置,无需再做改动。

(3)启动 minicom,按下目标板复位键,抢在目标板进入内核启动步骤之前按下回车键(否则目标板就进入由先前烧写进去的 Bootloader 引导的默认启动顺序),此后可以在超级终端上看到 blob>出现。在 blob>后键入 xdownload kernel 回车,告诉系统要下载的是内核,超级终端出现英语提示界面。键入 Ctrl+A 再键入 S(同前面的 Ctrl+A,Z,Ctrl+A 松开后才按 S),提示界面会变化,这时按键盘上下键选中 Xmodem 模式进行数据传输。回车即可。此时出现的界面是让你选定要烧到目标板上的文件。在屏幕下方有一排选择项,你可以用键盘左右键选中<Goto>,在出现的对话框中输入你文件所在的目录,回车,屏幕会列出你制定目录里的文件,用键盘上下键选中(箭头停在文件位置,按空格键选中该文件)你要烧的内核文件 zImage,回车就开始下载内核了。内核下载过程出现英语提示界面,下载完毕后,目标板发出一连串鸣声提示,到这一步内核程序只是被下载到 RAM 中,按回车后在提示符后键

入 flash kernel 后回车,这时目标板真正烧内核程序到 Flash 芯片中。

(4)这里值得注意的是,如果是一个不完全明白串口烧通信程序工作原理的新手,对写芯片的过程不熟,不能很好掌握串口通信程序的时序的情况,可能一直不能成功将程序写入芯片。

## 2.3 烧 romf 只读文件系统到 Flash

烧 romf 只读文件系统到 Flash 的步骤同 2.2 中的(1)、(2)和(4)是一样的。只是在(3)中有略微的不同。在如上的(3)中,当选需要烧到 Flash 中的文件时,这里要求选 romfs. bin 文件。最后当 romfs. bin 已经下载到 RAM 后,在提示符后键入 flash ramdisk 后回车。这时目标板真正烧文件系统到 Flash 芯片中。通过上述三个步骤, $\mu$ Clinux 操作系统真正移植完毕。按目标板复位键重新启动目标板,最后的启动在 minicom 中显示,可以看到启动界面:

```
.....
Command:mount -t ext2 /dev/ram1 /ramdisk
Command:chmod 777 /ramdisk
Command:mkdir /var/config
Command:mkdir /var/tmp
Command:mkdir /var/log
Command:cat /etc/motd
.....
```

Welcome to

```

      _ _ _
      /  _ | | |
      -  _ | | | _ _ _ _ _ _ _
      | | | | | | | _ | | | / /
      | | | | | | | | | | | | /
      |  _ _ | | | | | | | | /
      | |
      | |
      | |
```

For further information check:

<http://www.μClinux.org/>

Command:ifconfig lo 127.0.0.1

Command:route add -net 127.0.0.0 netmask 255.255.255.0 lo

Command:ifconfig eth0 192.168.253.2 netmask 255.255.255.0 up

Command:

Execution Finished, Exiting

Sash command shell (version1.1.1)

/>

这时,可以确认移植  $\mu$ Clinux 成功,可以在 minicom 中用 shell 命令来操作  $\mu$ Clinux 操作系统。进一步开发需要 mount 目标平台上的  $\mu$ Clinux 操作系统目录到宿主主机上,以方便应用程序的开发。

(下转第 84 页)

如图 2 所示,测试了两算法在相同评价指标下挖掘出的序列的准确率。在改进的算法中,挖掘出的频繁模式序列的准确率要比传统的 Prefixspan 算法挖掘出的序列的准确率高约 10%。

## 5 结束语

本研究将时间属性和频繁度的概念引入到 Prefixspan 算法中,提出了基于时间属性和点击量的 Prefixspan 改进算法。在改进的算法中,除了根据用户的访问序列进行挖掘以外,同时还将用户的访问时间和点击量因素加入到挖掘信息的过程中。提高了挖掘信息的实时性与准确性,从而使互联网时代的信息更加高效和准确地服务于用户。但是,该算法的改进是以部分时间和空间作为代价的,下一步的主要工作是如何进一步改进算法的时间复杂性和空间复杂性,提高算法的整体性能。

### 参考文献:

- [1] 孟宪苹,宋 菲,李 俊.基于序列模式挖掘的入侵检测系统的研究[J].计算机技术与发展,2008,18(3):154-156.
- [2] 叶和平,尚 敏.一种面向入侵检测的数据挖掘算法研究[J].计算机技术与发展,2008,18(11):149-151.
- [3] Agrawal R, Srikant R. Mining Sequential Patterns[C]//Proceedings of the 11th International Conference on Data Engineering. Taipei, Taiwan, China:IEEE Press,1995.
- [4] 吴 楠,胡学刚.基于聚类分区的序列模式挖掘算法研究[J].计算机技术与发展,2010,20(6):109-112.
- [5] 陈 伟.Apriori 算法的优化方法[J].计算机技术与发展,2009,19(6):80-83.
- [6] Srikanth R, Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements[R]. IBM Almaden Research Center,1995.
- [7] Chen Yen-Liang, Chinag Mei-Ching, Kao Ming-Yat. Discovering time-interval sequential in patterns databases [J]. Expert Systems with Application,2003,25(3):343-354.
- [8] 李锡娟,刘 峰,闫娜娜,等. Prefixspan 算法与 CloSpan 算法的分析与研究[J].计算机技术与发展,2008,18(1):70-73.
- [9] Tsai Chieh-Yuan, Chien Chun-Ju. An Optimized Classification Model for Time-Interval Sequences[C]//Proceedings of the World Congress on Engineering. London:[s. n.],2010.
- [10] 汪林林,范 军.基于 Prefixspan 的序列模式挖掘改进算法[J].计算机工程,2009,35(23):56-61.
- [11] Pei Jian, Han Jiawei, Mortazavi-Asl B, et al. Mining Sequential Patterns by Pattern-growth: The PrefixSpan Approach[J]. IEEE Trans. on Knowledge and Data Engineering, 2004, 16(11):1424-1440.
- [12] Han Jiawei, Pei Jian, Mortazavi-Asl B, et al. FreeSpan: Frequent Pattern-projected Sequential Pattern Mining[C]//Pro. of 2000 Int'l Conf. on Knowledge Discovery and Data Mining. New York, USA:ACM Press, 2000: 355-359.

(上接第 80 页)

## 3 结束语

介绍了开发嵌入式 Web 服务器环境的搭建和  $\mu$ Clinux 操作系统的移植,嵌入式系统的开发都是基于一定的硬件平台,如果对平台选取的合理,便于开发过程的控制。由于 Linux 和  $\mu$ Clinux 都是技术成熟的操作系统,并且源代码公开,因此课题的开发采用了这二种操作系统,并且详细地介绍了在 Redhat Linux 操作系统上搭建开发嵌入式 Web 服务器环境和在 NETARM3000 硬件平台  $\mu$ Clinux 操作系统的移植。只要完全理解论文的内容,相信可以很容易地设计一个适合多平台嵌入式操作系统的嵌入式 Web 服务器的开发环境。

### 参考文献:

- [1] 丁苍峰,薛宁静.嵌入式 Web 服务器身份认证的解决方案[J].计算机工程与设计,2008,29(24):6255-6257.
- [2] 朱玮玮,杨建明.  $\mu$ Clinux—一种嵌入式 Linux 系统[J].舰船电子工程,2003(4):56-59.
- [3] 沈 可.使用嵌入式 Linux 操作系统进行软件开发的特点及优势[J].电脑开发和应用,2001,14(7):13-15.
- [4] 徐 明.基于 S3C44B0 与  $\mu$ Clinux 构建 Web 服务器[J].可编程控制器和工厂自动化,2005(12):97-100.
- [5] 陈坚华,范 燕,冉立新,等.基于 ARM7TDMI 的  $\mu$ Clinux 移植[J].电子器件,2003,26(1):15-18.
- [6] 英海燕.基于 ARM 的嵌入式 Linux 操作系统移植[J].现代情报学,2005,25(5):155-156.
- [7] 王瑞恒.嵌入式 Internet 在 ARM 上的实现方法和研究[D].哈尔滨:哈尔滨工程大学,2008.
- [8] 郑音飞,胡大可.基于  $\mu$ Clinux 的嵌入式 Web 服务器的实现[J].电子产品世界,2003(7):50-51.
- [9] 赵 颖,周春光,吉 斌,等.嵌入式系统有限资源下的代码移植和优化[J].吉林大学学报(理学版),2007,45(6):995-998.
- [10] 袁爱君.基于 ARM 的嵌入式  $\mu$ Clinux 系统设计与 Web 服务器应用的实现[D].杭州:浙江大学,2003.
- [11] Venkateswaran S. Embedded bootloaders[J]. Embedded Systems Europe, 2004,8(55):32-34.
- [12] 孟 雷,忽海娜. ARM—Linux 嵌入式系统 BootLoader 的配置与移植[J].计算机技术与发展,2008,18(10):204-206.