基于S3C44B0X的U-Boot及μClinux的移植分析

郭建磊,杨厚俊
(青岛大学 信息工程学院,山东 青岛 266071)

摘 要:选取了当前比较流行、功能强大的引导程序U-Boot及专门应用于无MMU微处理器的μClinux操作系统,深入研究了U-Boot及μClinux的移植方法。以UP-NETARM3000开发板为例,详细分析其特殊功能寄存器设置及移植过程,创建了基于S3C44B0X的ARM-μClinux开发平台。在嵌入式系统的开发调试阶段,充分利用U-Boot网络引导方式大大提高了开发效率,为基于μClinux进行各种设备驱动程序的开发提供了便利。用文中方法移植的U-Boot及μClinux已稳定运行在UP-NETARM3000开发板上,为后续的嵌入式产品开发打下坚实的基础。

关键词:S3C44B0X;U-Boot;μClinux;特殊功能寄存器
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2009)04-0013-04

Transplant of U-Boot and μClinux Based on S3C44B0X

GUO Jian-lei, YANG Hou-jun
(Dept. of Information Engineering, Qingdao University, Qingdao 266071, China)

Abstract: Select U-Boot and μClinux to make an in-depth study of the method of transplanting, because U-Boot is popular and powerful now as bootloader and μClinux is special designed for MCU without MMU. Take UP-NETARM3000 exploring board for instance, analyze configuration of its special function register and the process of transplanting in detail. Construct an ARM-μClinux based embedded system production platform based on S3C44B0X. U-Boot and μClinux transplanted in this way run steadily on UP-NETARM3000 exploring board and make solid foundation for the following developing.

Key words: S3C44B0X; U-Boot; μClinux; special function register

0 引 言

S3C44B0X是三星公司推出的一款为手持设备或其他通用设备而开发的16/32位处理器,它基于ARM7TDMI[1]核,没有内存管理单元(MMU)。μClinux[2]是专门应用于无MMU微处理器的linux衍生操作系统,具有支持多任务、内核精简、高效、稳定等优点。bootloader[3]选取了当前比较流行的、功能强大的U-Boot,它可以支持多种体系结构的处理器,同时提供了完备的命令体系,在嵌入式系统的开发调试阶段,利用U-Boot的网络引导方式可以大大提高开发效率。

文中以博创UP-NETARM3000开发板为例,详细分析了基于S3C44B0X的U-Boot及μClinux的移植过程。

收稿日期:2008-08-04
基金项目:国家高技术研究发展计划(863计划)(2006AA01Z110)
作者简介:郭建磊(1984-),女,山东临沂人,硕士研究生,研究方向为嵌入式系统设计;杨厚俊,硕士,副教授,研究方向为计算机体系结构等。

1 U-Boot的移植

1.1 开发板的配置

UP-NETARM3000是由博创公司生产的嵌入式教学科研试验系统,采用三星S3C44B0X微处理器,工作频率为66MHz。2MB NOR Flash(AM29LV160D)作为引导ROM;8MB SDRAM(HY57V641620HGT);16MB NAND Flash(K29F2808U);基于RTL8019芯片的10M以太网接口;

目标板上的存储分配地址为:

NOR Flash地址:0x00000000-0x00200000

NAND Flash地址:0x02000000-0x03000000

SDRAM地址:0x0c000000-0x0c8000000

1.2 U-Boot移植

在移植过程中,根据UP-NETARM3000的硬件配置,选定同样基于S3C44B0X处理器的B2板作为参考。具体步骤如下:

(1) 修改U-Boot/Makefile文件。

在顶层Makefile中为开发板添加新的配置选项,如取名为myboard,在Makefile中添加以下两行:

myboard_config : unconfig

@./mkconfig $(@:_config=) arm s3c44b0 myboard

(2) 在 board 下创建主板目录。

在 board 下建立 myboard 目录,并将 board/dave/B2 下的文件拷贝到 myboard 下。此目录下的文件主要有:B2.c、flash.c、lowlevel_init.S、U-Boot.lds 等。将 B2.c 改名为 myboard.c。

(3) 创建头文件。

在 U-Boot/include/configs 目录下创建 myboard.h,并将 B2.h 的内容拷贝过来。

1.3 参数修改及特殊功能寄存器的配置

根据开发板硬件情况设置相关的参数及特殊功能寄存器,移植过程中需要修改的主要有以下几项:

1.3.1 修改 include/configs/myboard.h

此文件是关于目标板配置的一些宏定义,作以下修改:

```
#define CONFIG_S3C44B0_CLOCK_SPEED 66
#define CONFIG_BAUDRATE   115200
#define PHYS_SDRAM_1      0x0c000000
#define PHYS_SDRAM_1_SIZE 0x00800000
#define PHYS_FLASH_SIZE   0x00200000
#define CONFIG_DRIVER_RTL8019
#define RTL8019_BASE      0x0a000600
```

1.3.2 修改 board/myboard 目录下的文件

该目录下需要修改的文件有 myboard.c 和 lowlevel_init.S

(1) myboard.c:该文件主要包括微处理器各个 I/O 端口的初始化以及 SDRAM 的驱动程序及初始化。根据数据手册修改此文件中关于微处理器各个 I/O 端口的初始化部分。

(2) 修改 lowlevel_init.S:该文件是汇编程序,包括 SDRAM 的工作参数设置以及处理器内存控制器的初始化。

根据数据手册及开发板配置可做如下初始化[4]:

```
MEMORY_CONFIG:
.long 0x11110002 /* 总线宽度和等待状态寄存器,BANK7、6、5、4、0 数据宽度为 16 位, BANK3、2、1 为 8 位,模式为小端 */
.word 0x00000600 /* GCS0 ,boot ROM, Tacc = 10clk */
.word 0x00007ffc /* GCS1 , K9F2808 (三星 16Mbyte Flash),非线性寻址 */
.word 0x00007ffc /* GCS2 ,USBN9603,USB 设备端接口芯片,占用系统外部中断 0。8 位数据总线 */
.word 0x00007ffc /* GCS3 ,未接设备,可以供扩展使用 */
.word 0x00007ffc /* GCS4 ,未接设备,可以供扩展使用 */
.word 0x00007ffc /* GCS5 ,RTL8019AS,ISA 总线兼容的 10M 以太网(PHY + MAC 层)控制芯片,占用系统外部中断 1,16 位数据总线 */
.word 0x00018000 /* GCS6,SDRAM,(Trcd=2, SCAN=8) */
.word 0x00018000 /* GCS7 ,未接设备,可以供扩展使用 */
.word 0x009603fb /* REFRESH 刷新控制寄存器的配置 */
.word 0x16 /* SCLK power down mode, BANKSIZE 8M */
.word 0x20 /* MRSR6 CL=2clk */
.word 0x20 /* MRSR7 CL=2clk */
```

1.3.3 修改 cpu/s3c44b0 目录下的文件

该目录下需要修改的文件有 start.S 和 serial.c:

(1)start.S:该文件是 U-Boot 启动时的入口代码,其完成的主要工作有:禁止看门狗、禁止中断、设置微处理器工作频率、设置 PLL、设置中断向量等。

目标板的外部晶振为 6MHz,根据公式[5]:

$f_{pllo}=(m*f_{in})/(p*2^s)$,m=MDIV+8, p=PDIV+2, s=SDIV

计算出 PLLCON,并作如下修改:

```
#if CONFIG_S3C44B0_CLOCK_SPEED=66
ldr r0, =0x7c041
```

(2)serial.c:此文件主要完成串口驱动的初始化。根据以下公式 计算分频因子[5]: UBRDIVn = (int)(MCLK/(波特率 * 16)) - 1

并在源代码中作如下修改:

```
case 115200:
#if CONFIG_S3C44B0_CLOCK_SPEED=66
divisor=35;
```

1.4 编译代码,烧写到 flash

代码修改完成后,重新进行编译。分别执行 make myboard_config, make,将编译后生成的 uboot.bin 文件通过 JTAG 口烧写到 flash 的 0x0 处。在宿主机上安装 tftp 软件包后,即可以通过 U-Boot 的 tftp 命令下载内核映像及根文件系统到开发板上。开发板上电时 U-Boot 启动[6],可以在超级终端[7]看到串口打印出的消息。如图 1 所示。

```
U-Boot 1.1.4 (Jul 21 2008 - 09:45:38)

U-Boot code: 0C700000 -> 0C71B100  BSS: -> 0C71F6CC
RAM Configuration:
Bank #0: 0c000000  8 MB
Flash:  2 MB
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  0
=>  tftp c008000 image.ram
TFTP from server 10.1.58.200; our IP address is 10.1.58.56
Filename 'image.ram'.
Load address: 0xc008000
Loading: ##################################################
         ##################################################
         ##################################################
         ##################################################
         ##################################################
         ##################################
done
Bytes transferred = 1847280 (1c2ff0 hex)
=>
```

图 1　U-Boot 启动下载信息

2　μClinux 的移植过程

2.1　移植准备

需要准备的软件包[8]:

系统包:μClinux-dist-20040408.tar.gz

补丁包:μClinux-20040408-ARMSYS.patch

补丁包的作用是针对 S3C44B0X 对系统进行配置。

2.2　移植过程

(1)解压 μClinux-dist-20040408.tar.gz

(2)创建 μClinux-dist/vendors/Samsung/44B0

(3)将 4510 目录下的文件拷贝到 44B0 目录下;

(4)打补丁:将补丁文件拷贝到 μClinux-dist 下,执行如下命令打补丁: patch -p1 < μClinux-20040408-ARMSYS.patch

当补丁文件执行到 patching file vendors/Samsung/44B0/Makefile 时会出现错误,以下内容将对照补丁进行手动修改。

2.3　对照补丁修改文件

打开 μClinux-20040408-ARMSYS.patch,找到 diff -Naur μClinux-dist/vendors/Samsung/44B0/Makefile,对照此文件修改 μClinux-dist 中的各相应文件。

(1)修改/vendors/Samsung/44B0/Makefile 文件。

(2)修改 μClinux-dist/vendors/Samsung/44B0/rc 增加以下几行:

```
/bin/expand /etc/ramfs2048.img /dev/ramdisk
mount -t ext2 /dev/ram1 /var
chmod 777 /ramdisk
ifconfig eth0 192.168.1.76 netmask 255.255.255.0 up
```

(3)修改 μClinux-dist/linux-2.4.x/drivers/block/blkmem.c

添加 #ifdef CONFIG_BOARD_MBA44

extern char romfs_data[];

extern char romfs_data_end[];

#endif

将 #ifdef CONFIG_BOARD_MBA44

{0, 0XC7000000, -1} 改为:

#ifdef CONFIG_BOARD_MBA44

{0, romfs_data, -1}

(4)修改 μClinux-dist/linux-2.4.x/drivers/char/Makefile

添加:

obj - $(CONFIG_SERIAL_S3C44B0X) += serial_S3C44B0X.o

(5)修改 μClinux-dist/linux-2.4.x/arch/armnommu/config.in

添加:define_bool CONFIG_SERIAL_S3C44B0X

(6)增加 bzImage

修改 μClinux-dist/Makefile

在 TFTPDIR = /tftpboot 后添加

LINUXTARGET = bzImage

(7)更改内核在 sdram 的运行地址

修改 μClinux-dist/linux-2.4.x/arch/armnommu/boot/Makefile:

```
ifeq( $(CONFIG_ARCH_MBA44),y)
ZTEXTADDR = 0x0c008000
/* 压缩内核自解压代码的起始地址[9] */
ZREALADDR = 0x0c300000
/* 内核解压后代码输出的起始地址[9] */
```

修改 μClinux-dist/linux-2.4.x/arch/armnommu/Makefile:

```
ifeq( $(CONFIG_ARCH_MBA44),y)
TEXTADDR = 0x0c008000
/* 非压缩内核起始运行地址[9] */
```

2.4　设置开发板信息

2.4.1　修改串口通信波特率

打开 μClinux-dist/vendors/Samsung/44B0/config.arch,将波特率改为 U-Boot 统一的波特率:

CONSOLE_BAUD_RATE=115200

2.4.2　修改 μClinux-dist/vendors/Samsung/44B0/config.linux-2.4.x

主要修改的几点:

选择开发板:CONFIG_BOARD_MBA44=y

修改 CPU 型号:CONFIG_CPU_S3C44B0X=y

选择内核模式:CONFIG_RAMKERNEL=y

设置 DRAM 和 FLASH 参数。

2.4.3 修改 μClinux - dist/vendors/Samsung/44B0/config.vendor - 2.4.x

(1)注释掉:CONFIG_USER_BUSYBOX_KILLALL=y

(2)将 CONFIG_USER_LOGIN_LOGIN=y 改为 CONFIG_USER_LOGIN_LOGIN=n

(3)将 CONFIG_USER_OLD_PASSWORD=y 改为 CONFIG_USER_OLD_PASSWORD=n

(4)将 CONFIG_USER_BUSYBOX_BUSYBOX=y 改为 CONFIG_USER_BUSYBOX_BUSYBOX=n

3 编译内核

在 μClinux - dist 目录下依次执行以下命令编译内核:make distclean; make menuconfig; make dep; make lib_only;make user_only;make romfs;make linux;make image ;make。

编译通过后会在 μClinux - dist/images 里产生三个文件:

image.ram image.rom romfs.img

将生成的 image.ram 通过 U - Boot 的 tftp 命令下载到内存的 0xc008000 处,用 go 命令运行即可看见运行结果。

4 结束语

介绍了基于 S3C44B0X[10] 的 U - Boot 及 μClinux 的移植过程,并以 UP - NETARM3000 开发板为例详细分析了特殊功能寄存器的配置过程及移植步骤。通过文中方法建立的 ARM - μClinux 开发平台已经稳定运行在笔者的开发板上,为后续嵌入式开发打下了坚实的基础。

参考文献:

[1] 李 岩, 荣盘祥. 基于 S3C44B0X 嵌入式 μClinux 系统原理及应用[M]. 北京:清华大学出版社,2005.

[2] Chou Chi - Hung, Yang Tsung - Hsien, Tsao Shih - Chiang, et al. Standard opeating procedures for embeded linux systems [J]. Linux Journal, 2007(160):10 - 14.

[3] 王金柱,卢 迪,张开钰. ARM7 的 bootloader 浅析[J]. 自动化技术与应用,2008,27(2):124 - 125.

[4] Samsung Company. S3C44B0X datasheet[DB/OL]. 2004. http://www.hyesco.com/download/ARM/s3c44b0x.rar.

[5] 田 泽. 嵌入式系统开发与应用[M]. 北京:北京航空航天大学出版社,2005.

[6] 陈海军, 申卫昌, 史 颖. 嵌入式系统引导程序详探[J]. 计算机技术与发展,2006,16(1):123 - 125.

[7] ARM Software Development Toolkit Version 2.0 - Programming Techniques[DB/OL]. 2004. http://infoeng.ee.ic.ac.uk/gac1/Architecture/Progtech.pdf.

[8] 高 卓. 基于 ARM 的 μClinux 移植与开发[J]. 微计算机信息,2007,23(8 - 2):151 - 153.

[9] 孙天泽,袁文菊. 嵌入式设计及 linux 驱动开发指南——基于 ARM9 处理器[M]. 第 2 版. 北京:电子工业出版社,2007.

[10] 李江乐,宗 容,裴以建,等. S3C44B0X 的 uC/OS - Ⅱ移植技术研究[J]. 计算机技术与发展,2008,18(8):134 - 136.

(上接第 12 页)

中的算法。约简过程在依赖度函数不减小的情况下进行,做到了有效的"无损压缩",使得知识的规则表示尽可能的简洁,试验表明依赖度函数值的计算开销比较小。这里只求出一个约简,避免了很多不必要的开销。

4 结束语

主要研究粗糙集的属性约简问题,提出基于禁忌搜索的 TSAR 方法,其中的广泛性和集中性机制是有效的,数值试验中和其他的方法相比,此方法是有前景的,依赖度函数的计算开销比较低。所以,禁忌搜索成功地用于粗糙集的属性约简问题,获得了合理的解,且在节省开销方面有高超的表现。实际的应用系统中,用户一般选择低开销的方法,即使他们只获得了次最优的结果。最后,这个方法拥有太多的参数,然而,这样在试验初始阶段可以调整参数,使其更智能化,进而试验结果与初始参数的选择不敏感。

参考文献:

[1] Pawlak Z. Rough Sets - theoretical Aspects of Reasoning about Date[M]. Dordrcht:Kluwer Acadamic Publisher,1991.

[2] 刘银山,吴孟达,王 丹. 知识发现中属性约简算法的研究[C]//中国企业运筹学学术交流大会论文集. 成都:电子科技大学出版社,2005:128 - 132.

[3] 张鸿宾,孙广煜. Tabu 搜索在特征选择中的应用[J]. 自动化学报,1999,25(4):457 - 466.

[4] 刘 清. Rough 集及 Rough 推理[M]. 北京:科学出版社,2001.

[5] Glover F, Laguna M. Tabu Search[M]. Boston:Kluwer Academic Publishers, 1997.

[6] 徐 燕,李锦涛,王 斌,等. 一种新颖的基于粗糙集的特征选择方法[C]//中国计算技术与语言问题研究——第七届中文信息处理国际会议论文集. 北京:电子工业出版社,2007:480 - 487.