

嵌入式 Linux 在车载导航系统中的应用

潘成源, 徐 勇, 李 鑫

(湖南大学 电气与信息工程学院, 湖南 长沙 410082)

摘 要: 描述了将 Linux 移植到基于 MOTOROLA MX1 处理器的目标板上的方法和过程。介绍了导航系统的原理, 分析了嵌入式 Linux 操作系统及文件系统, 并结合导航系统介绍了嵌入式 Linux 的具体开发步骤, 着重介绍 Linux 内核的移植。结果证明该方法是可行的。移植后的 Linux 在目标板中运行良好。

关键词: 嵌入式 Linux; 文件系统; 驱动程序; 导航系统

中图分类号: TP316.8

文献标识码: A

文章编号: 1673-629X(2006)05-0223-03

Embedded Linux and Its Application in Vehicle Navigation System

PAN Cheng-yuan, XU Yong, LI Xin

(College of Electrical and Information Engineering, Hunan University, Changsha 410082, China)

Abstract: The method and process of porting Linux to the board based on MOTOROLA MX1 processor are described. Firstly, the schematic drawing of system is analysed. Meanwhile, the embedded Linux operation system is analysed as well as file system. Steps of development of embedded Linux in vehicle navigation are detailed in this paper. The porting of Linux kernel are discussed in detail. The porting Linux works well in the target board.

Key words: embedded Linux; file system; driver; navigation system

近年来,在国内对嵌入式 Linux 的研究可谓方兴未艾,越来越多的产品开发采用嵌入式 Linux 作为其操作系统。究其原因在于其价格低廉,尤其是它的内核公开源代码,这是很多商用嵌入式操作系统所没法比拟的优势,并有着很好的注释和文档说明,可以清楚地把握操作系统的运行机理,可针对具体的应用在内核中的源码进行优化。

1 硬件平台介绍

车载导航系统是用于汽车上并辅助驾驶的智能电子设备,它和一般的电子设备一样,也是由硬件和软件两部分构成,本车载导航系统的硬件构成如图 1 所示^[1]。

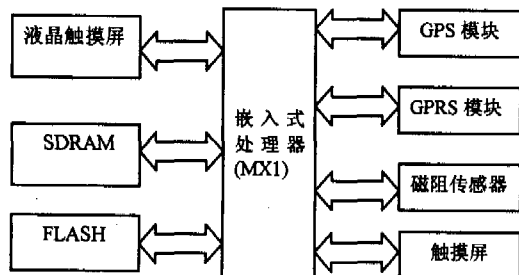


图 1 车载导航系统硬件框图

其中包括嵌入式处理器、SDRAM、FLASH、LCD、触摸屏、GPS 模块和 GPRS 模块等部分,嵌入式处理器选用了摩托罗拉公司龙珠系列的 MX1,它是基于 ARM920T 核的 RISC 处理器,其主频可高达 200MHz^[2],目标板上提供 32MFLASH 和 64MSDRAM,可满足导航系统的计算及图形显示对存储空间的要求,并采用 640×480 的真彩 LCD 屏,满足娱乐需要。此外,采用 GPRS 模块进行通讯,可以实时监控车辆并进行报警。

2 软件构成

导航系统软件大致分为 5 个部分:引导装载程序(Bootloader)、内核(kernel)、文件系统(File system)、图形用户界面(GUI)和上层应用程序(Application)。其中内核和文件系统构成系统的软件平台,是系统开发中的两大关键技术。文中将着重论述如何将 Linux 移植到基于 MX1 处理器的开发平台上,特别是驱动程序的编写以及文件系统的制作。

3 基于 MX1 的嵌入式 Linux 系统开发

当前,有很多稳定完善的商用操作系统,如:Vx-works, PalmOS, WINCE, Nucleus, QNX 等,但其高昂的费用令许多小厂家望而却步,而 Linux 虽非商业操作系统,但也很稳定,功能强大又易于移植而正在被广泛采用。Linux 操作系统采用一体化内核,包括进程管理、存储管

收稿日期:2005-11-14

作者简介:潘成源(1976-),男,广西人,硕士研究生,研究方向为嵌入式系统、车载导航等;徐 勇,副教授,硕士生导师,主要研究方向是计算机测控系统、嵌入式系统。

理、文件系统管理、设备管理以及网络通信等部分。Linux 最初是以 X86 体系结构为蓝本而设计的,后来人们又把它移植到其他一些处理器,比如:POWERPC, MIPS, alpha, ARM 等。把标准的 Linux 用于 ARM,显然是不完全适合的,需要在标准版本上进行修改、扩充、移植,让其适合于特定处理器和存储器。要让 Linux 运行在自己的开发板上,首先要建立交叉编译环境。

3.1 建立交叉编译环境

所谓交叉编译,简单地讲就是在一个平台上生成可以在另一个平台上执行的代码。就本系统而言就是在 X86 机器上编译运行于 ARM 的程序^[3]。Linux 环境下的 GNU 交叉编译工具包主要包括 GNU 编译器 gcc、实用工具 BINUTILS(包括 as, ld, objcopy, readelf 等)和运行库 glibc。有两种方法建立交叉编译环境:其一是从网上下载已经编译好的二进制代码,但需注意下载的版本是否已支持 MX1;二是下载源代码自己生成二进制代码,这包括下载源代码,修补补丁,配置、编译、设置头文件及安装等操作,源代码的下载可以到 GUN 官方网站去下载,配置时指明 CPU 为 ARM,并设好安装路径,而后就可以运行 make 进行编译以及 make install 安装了。

3.2 Linux 内核的移植与裁减

标准版本 Linux 内核也支持 ARM 处理器,但作为面向嵌入式系统的 ARM 中有一些资源和功能在原版 Linux 中是用不到的,或者不存在,而对于嵌入式系统却意义重大,因此需要在原版的 Linux 上进行修改、扩充和移植,主要包括以下几个部分:首先是 CPU core 的移植,对 MX1 而言其核是 ARM920T;其次是 Linux 系统内核对片上系统(SOC)及平台的支持;最后是针对外部设备编写相应的驱动程序,如液晶显示器、触摸屏、键盘等。对于 MX1,其采用的是 ARM 内核的处理器,需要修改移植的源代码分布:头文件分布在 Linux/include/asm-arm;与 CPU core、片上系统及平台有关的源代码分布在 Linux/arch/arm;外部设备的驱动程序则分布在 Linux/drivers 下。

首先要修改或者重新编写的是与 CPU core 相关的内核代码,主要包括内存管理及高速缓存的锁定,由于 ARM 体系架构和 X86 不同,所以它的内存管理机制也有所不同,虽然也是采用分段和分页管理,但页面大小并不完全一样,因此需要改动内存头文件。此外,嵌入式系统中的内存的组成与布局比通用计算机系统复杂、多样,因此需要一个“机型描述块”,即 machine_desc 数据结构进行描述,对于具体机型的描述块,通过条件编译来选用;在此基础上,还需要有“映射描述块”作为补充,即 map_desc 数据结构,用来描述内存地址一个区间的映射,包括其虚拟地址、物理地址、区间的长度、所属的域、是否可读/可写、是否可以高速缓存、是否可以写缓冲。至于高速缓存的锁定,主要包括对 Dcache, TLB, Icache, MMU 的设计等,这些代码一般由硬件厂商提供,仅需作少量修改。其次是对片上系统及开发平台的支持,这部分主要包括对中断控制

器、时钟控制器支持与初始化, CPU 频率的设定, IO 端口的映射等。虽然各种体系架构的中断响应框架基本相同,但其实现却又与硬件及系统结构有密切的联系,所以这部分代码也要重写,这部分移植的工作量较大,可参考相关类似的平台来改写^[4]。此外,ARM-Linux 的系统调用也和原版 Linux 的系统调用有所不同。最后就是系统外围设备驱动程序的编写,包括 LCD、触摸屏、串口、SPI 口、I2C 总线等的驱动,这部分工作量非常大,是系统开发的关键,因此着重介绍。Linux 系统的设备分为以下三种:字符设备、块设备和网络设备。设备驱动程序是操作系统内核与机器硬件之间的接口,为应用程序屏蔽了硬件的细节,在 Linux 中,应用程序可以像操作普通文件一样对硬件设备进行操作,其主要由以下三部分组成:初始化子程序,服务于 I/O 请求的子程序及中断服务子程序。可以在硬件生产厂家或者 Internet 上寻找驱动程序,找不到则根据相近的驱动程序来改写,这样可以加快开发进度。嵌入式 Linux 驱动的开发流程如下^[5]:

- (1) 定义主、次设备号,也可以动态获取。
- (2) 实现初始化和清除函数。若采用模块方式,则要实现模块的初始化和清除函数。
- (3) 设计所要实现的文件操作,定义 file_operations 结构。
- (4) 实现所需要的文件操作调用,如 read, write 等。
- (5) 实现中断服务函数,用 request_irq 向内核注册。
- (6) 将驱动编译到内核或者编译成模块,用 ismod 命令加载。
- (7) 生成设备节点文件。

完成以上三部分代码的移植后,下一步需要修改的就是与内核启动有关的代码。Linux 的入口是一段汇编代码,用于基本的硬件设置和建立临时页表,对于 ARM-Linux 是 head-armv.S,连接时它位于内核代码的最前面。在 head-armv.S 中需要注意的是:由于不是从 Bootloader 中把当前 machine 的 architecture number 传递下来,所以这里需要手工设置。内核开始从 ENTRY(stext)开始执行:设置 CPSR 状态,判断 processor 和 architecture 类型,创建核心页表,跳转到 CPU 初始化程序,返回 -ret 处,更新 cp15 寄存器,打开 MMU, Cache, buffer 和系统保护,跳转至 -mmap-s-watched 处执行,清 BSS,初始化 sp,保存与系统相关信息,最后进入 start_kernel 函数,真正开始内核初始化的工作,先初始化与平台相关的构架,然后再初始化陷阱、中断、进程调度、定时器、控制台等等。其中与平台相关的初始化是移植主要关注的部分:其函数为 setup_arch,命令行字符串和内存起始地址作为参数传递到这个函数,这个函数首先初始化 CPU,然后再初始化平台,在 mdesc = setup_architecture(machine_arch_type)中用到以下重要数据结构:

```
MACHINE_START(MX1ADS, "Motorola MX1ADS")
MAINTAINER("WESG SPS Motorola")
```

```
# ifdef CONFIG_ARCH_MX1ADS_SRAM
    BOOT_MEM(0x12000000, 0x00200000, 0xf0200000)
# else
    BOOT_MEM(0x08000000, 0x00200000, 0xf0200000)
# endif

    FIXUP(mx1ads_fixup)
    MAPIO(mx1ads_map_io)
    INITIRQ(mx1ads_init_irq)
MACHINE_END
```

通过这些宏操作,设置了开发板中 RAM 的物理地址的起点,以及用于 I/O 空间物理地址和 I/O 虚拟地址的起点。注册用于 MX1 及其开发平台的 fixup 函数、I/O map 函数和 IQR 初始化函数。

到此完成了移植的大部分工作,但还有少量工作需要完成,如把设备驱动加到内核中需要修改相关目录下的 makefile 和 config.in 文件。makefile 定义 Linux 内核的编译规则,决定哪些文件能被编译到内核中,config.in 则给用户提供了配置选择的功能,决定有哪些配置选项。最后还需要根据系统的设置修改 Linux/arch/arm 目录下的内核链接脚本文件 vm-linux.lds,这个链接脚本文件定义各个模块的装载地址。至此,内核的移植工作基本完成。接下来要做的工作是对内核进行裁减和配置,首先是要选择系统运行所必需的一些模块,比如相关处理器类型及板级支持,然后根据系统所支持的外部设备来选择相应的选项,如 MTD 设备支持,不需要 ATA 硬盘支持等。当然对于一些外部设备可以编译成模块形式动态加载。至于文件系统模块及网络模块的支持,满足系统需要即可,以便节省系统存储空间,本系统选择支持 cramfs 及 RAMDISK。为了便于调试及下载,增加以太网的支持。最后运行一下编译命令即可产生所需的二进制映像,位于 /linux/arch/arm/boot/目录下,可以通过 Bootloader 或 JTAG 口烧入开发板的闪存中。

4 准备根文件系统

嵌入式 Linux 系统的运行除了内核映像外还需要用户空间的管理程序、配置文件、启动脚本文件及运行库等的支持,因此需要创建一个内核启动后可以使用的根文件系统。此外,嵌入式系统由于受到本身体积所限,一般不使用硬盘,而使用 Flash 来存储文件,但与硬盘类似,需要在 Flash 上创建文件系统才可以被 Linux 识别和使用。虽然 Linux 支持绝大多数文件系统,但在本系统中选用

cramfs,原因在于这种操作系统是只读的,有利于保护内核等重要内容不被意外修改,而且启动速度较快。为了创建文件系统,可以在 PC 机的 Linux 环境下建立一个目录,向目录中拷贝需要出现在开发板的 Linux 系统中的文件和目录,如目录:bin,etc,lib,dev 等;基本的工具:sh,ls,cp,mv 等;必需的配置文件:inittab,rc,fstab 等;必需的设备:/dev/tty*,/dev/console,/dev/mem 等等;还有必需的运行库:glibc。然后就可以使用工具 mkcramfs 来生成镜像文件,最后使用 Bootloader 将镜像烧到 Flash 中特定地址即可被内核识别和使用。

由于 cramfs 是只读的,还需要写一个操作 RAM 的驱动程序,伪装成硬盘,并且挂载到/tmp 目录,因此可写。

5 结束语

主要介绍了基于嵌入式应用处理器 MX1 的 Linux 系统的移植,移植过程中,笔者得出以下体会:

(1)在移植之前,需要详细了解目标平台的系统结构、CPU 的体系结构等硬件知识;

(2)移植时,要讲究策略:先实现最基本的功能,然后再扩展其他功能;

(3)Linux 是自由软件,依赖于国际上很多工程师的支持。因此需多留意 Internet 网上的资源,特别是 E-MAIL 列表,全世界的开发者都可以在上面探讨问题。

导航系统采用嵌入式 Linux 作为操作系统,将会大大降低软件成本,提高系统的安全性。同时,嵌入式系统是国家“十五”发展的重点方向,越来越多的厂商开始采用嵌入式 Linux 作为其产品的操作系统,因而对嵌入式 Linux 的研究具有现实意义。

参考文献:

- [1] 谭磊. 基于嵌入式 Linux 的智能手机系统设计[J]. 计算机应用, 2004(12): 4-6.
- [2] MC9328MX1 i. MX Integrated Portable System Processor Reference Manual[Z]. Freescale Semiconductor, Inc, 2004.
- [3] 张杰,曹卫华,吴敏,等. 基于 S3C2410 的 Linux 移植[J]. 微机发展, 2005, 15(6): 142-144.
- [4] 毛德操,胡希明. 嵌入式系统——采用公开源代码和 StrongARM/XScale 处理器[M]. 杭州: 浙江大学出版社, 2003.
- [5] 马忠梅,李善平,康慨,等. ARM&Linux 嵌入式系统[M]. 北京: 北京航空航天大学出版社, 2004.

(上接第 130 页)

- proach to Change Detection: Algorithms, Analysis and Application[A]. Proc IEEE Symposium on Research in Security and privacy[C]. Oakland, CA: [s. n.], 1996. 110-119.
- [7] Janeway C A, Travers P. Immunobiology: The Immune System in Health and Disease, (2nd ed)[M]. London: Current Biology Ltd., 1996.

- [8] Perelson A S. Theoretical Immunology[M]. [s. l.]: Addison-Wesley, 1988.
- [9] Ayara M, Timmis J, Lemos L, et al. Negative Selection: How to Generate Detectors[A]. Proceedings of the 1st International Conference on Artificial Immune Systems(ICARIS)[C]. Canterbury, UK: [s. n.], 2002. 89-98.