

嵌入式平台上的网络通信模块驱动的实现

Implementation of the Drivers of the Network Communication Modules on Embedded Platform

蔡立三 李霖野 张红梅 (中国科学院空间科学与应用研究中心 北京 100080;
中国科学院研究生院 北京 100049)

摘要: 本文分析了嵌入式 Linux 系统下的网络通信模块驱动的实现原理,并具体通过 DM9000 以太网卡说明了其实现过程。

关键词: 嵌入式 Linux 网络 驱动 dm9000

目前, Linux 操作系统已经越来越广泛的被应用于嵌入式设备,这主要是由于它具有良好的开放性,并且其上的应用软件的源代码也是完全公开的。Linux 的网络设备驱动程序是 Linux 操作系统内核的一个重要组成部分,并且在嵌入式设备上有着重要的应用。本文重点对 Linux 网络设备驱动程序进行了分析,并具体通过嵌入式系统下的 dm9000^[1]以太网口,介绍嵌入式设备上的 Linux 系统的网络接口驱动的实现过程。

1 Linux系统下网络驱动程序的简介^[2]

在 Linux 中,为了简化对设备的管理,将所有的外围设备都归结为三类:字符设备(如键盘,鼠标等)、块设备(如硬盘、软驱等)和网络设备(如网卡、串口等)。为了将网络环境中的物理网络设备的多样性屏蔽, Linux 对所有的网络物理设备抽象并且定义了一个统一的概念:接口(Interface)。对于所有的网络硬件都是通过接口进行访问的,接口实际上提供了一个对于所有类型的网络硬件的一致化的操作集合,用于处理对数据的发送和接收。对于每一个已经驱动了的网络设备,都用一个 struct device 的数据结构表示。在内核启动或者驱动模块插入时,通过网络驱动程序,向系统注册检测到的网络设备。在进行网络数据传输的时候,网络驱动程序通过标准的接口将数据发送到相应的网络层,或者向网络发送数据包。

由数据结构 struct device 来表示网络设备在内

核中的运行情况,即网络设备接口,它既包括纯软件网络设备接口,如环路(Loopback),也可以包括硬件网络设备接口,如以太网卡。而由以 dev_base 为头指针的设备链表来集体管理所有网络设备,该设备链表中的每个元素代表一个网络设备接口。数据结构 device 中有很多供系统访问和协议层调用的设备方法,包括供设备初始化和往系统注册用的 init 函数,打开和关闭网络设备的 open 和 stop 函数,处理数据包发送的函数 hard_start_xmit,以及中断处理函数等。有关 device 数据结构(在内核中也就是 net_device)的详细内容,可以参看 /Linux/include/Linux/netdevice.h。

2 Linux网络设备驱动的体系结构

Linux 网络驱动程序的体系结构如图 1 所示。可以划分为四层,从上到下分别为网络协议接口层、网络设备接口层、提供实际功能的设备驱动功能层,以及网络设备和网络媒介层。

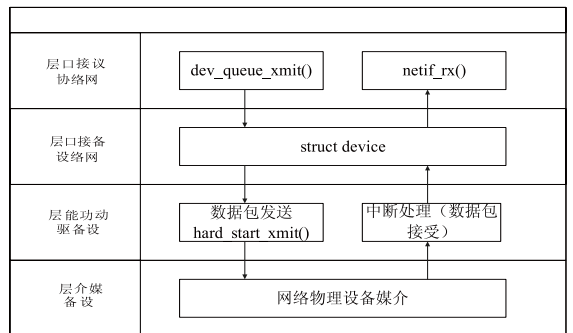


图 1 网络驱动程序的体系结构

Linux 内核源代码中提供了网络设备接口及以上层次的代码,因此移植特定网络硬件的驱动程序的主要工作就是完成设备驱动功能层的相应代码,根据底层具体的硬件特性,定义网络设备接口 struct net_device 类型的结构体变量,并实现其中相应的操作函数及中断处理程序。

struct net_device 结构体是整个网络驱动结构的核心,其中定义了很多供网络协议接口层调用设备的标准方法,net_device 结构保存和网络设备相关的任何信息。每一个网络设备都对应一个这样的结构,包括真实设备(例如以太网卡)和虚拟设备(比如 bonding 或 VLAN)。

3 网络驱动程序的编写及实现原理

Linux 网络系统各个层次之间的数据传送都是通过套接字缓冲区 sk_buf 完成的,sk_buf 数据结构是各层协议数据处理的对象。sk_buf<Linux/skbuff.h>是驱动程序与网络之间交换数据的媒介,驱动程序向网络发送数据时,必须从其中获取数据源和数据长度;驱动程序从网络上接收到数据后也要将数据保存到 sk_buf 中才能交给上层协议处理。

对于实际开发以太网驱动程序,可以参照内核源码树中的相应模板程序,重点理解网络驱动的实现原理和程序的结构框架,然后针对开发的特定硬件改写代码,实现相应的操作函数。下面以工作在 Linux 2.6.8.1 平台上的嵌入式 dm9000 网卡驱动程序^[3]为例,说明网络驱动程序的实现原理。

4 DM9000驱动的实现过程

4.1 设备初始化

网络的初始化是设备工作的第一步。当系统加载网络驱动模块的时候,就会调用初始化过程。首先利用函数 request_mem_region 映射 DM9000 的数据、地址端口,通过 dmfe_probe 函数检测网络物理设备是否存在,检测 DM9000 内部串行 NIC 的值是否正确,然后再对设备进行资源配置 构造设备的 net_device 数据结构。包括一些低层硬件信息 base_addr(网络接口的 I/O 基地址),IRQ(安排的中断号)等。

4.2 打开设备和关闭设备

open 方法在网络设备被激活的时候被调用,具体 DM9000 的硬件初始化工作放到这里来做。对于 DM9000 需要完成的初始化包括:对 DM9000 内部

上电,软件复位,通过 NCR 寄存器设置网络工作模式,可以选择设置内部或者外部 PHY、全双工或者半双工模式、使能唤醒事件等网络操作,对 RX/TX 中断使能,使能数据接收功能。调用 request_irq()申请中断号登记中断处理函数,调用 netif_carrier_on 侦测连接状态。启动定时器,调用 netif_start_queue 激活设备发送队列。

close 所做的工作和 open 相反,主要释放 open 获得的资源,以减少系统负担。

4.3 数据包的发送

数据包的发送和接收是实现 Linux 网络驱动程序中两个最关键的过程,对这两个过程处理的好坏将直接影响到驱动程序的整体运行质量。

数据传输通过 hard_start_xmit 函数实现,首先把存放在套接字缓冲区中的数据发送到网络芯片 DM9000 内部的 TX FIFO SRAM 中,该缓冲区是由数据结构 sk_buff 表示,把要传送的数据长度写入 DM9000 中的传输包长度寄存器 TXPLL 和 TXPLH 中。然后使能传输。如果数据发送成功,则会触发一次中断。

实际中会出现多个数据帧传输,需要考虑做并发处理。在发送时检测传输队列暂时满载时就要通过 netif_stop_queue 来暂停,当发送完成触发中断处理时,调用 netif_wake_queue 函数来重新启动传输队列。

网络传输由于系统忙或硬件的问题发生延迟,则会调用传送超时处理函数 tx_timeout,对硬件复位操作。

4.4 数据包的接收

数据包的接收是通过中断处理,当有数据到达时,就产生中断信号,网络设备驱动功能层就调用数据包接收程序来处理数据包的接收,然后网络协议接口层调用 netif_rx 函数把接收到的数据包传输到网络协议的上层进行处理。

DM9000 接收缓存区中的每帧数据由 4 字节长的首部、有效数据和 CRC 校验序列构成。首部 4 字节依次是 01H、以太网帧状态、以太网帧长度低字节和长度高字节,数据包接收程序首先检测如果第一个字节是 01H,则说明有数据;如果是 00H,则说明无数据,需要进行复位。然后调用函数 dev_alloc_skb 申请一块 sk_buff 结构缓冲区,从 DM9000 读出数据放置到缓冲区里,根据获取的长度信息,判断是否读完一帧。如果

读完,接着读下一帧,直到遇到首字节是 00H 的帧,说明接收数据已读完。接下来填充 sk_buff 中的一些信息,使之成为规范的 sk_buff 结构,最后调用 netif_rx()函数将接收到的数据传输到网络协议的上层。

4.5 驱动执行的流程

下图是 Linux 内核处理网络驱动的大致流程。首先,当系统探测到网卡时,系统会调用 insmod 命令来装载模块驱动,insmod 命令首先调用 init_module()函数来初始化模块,初始化完成后通过 register_netdev()来注册网络模块,这是模块的装载过程。接下来就是前面讨论的驱动的处理过程;最后是驱动结束后的卸载过程,先关闭网络接口设备,然后通过 rmmod 命令卸载模块。

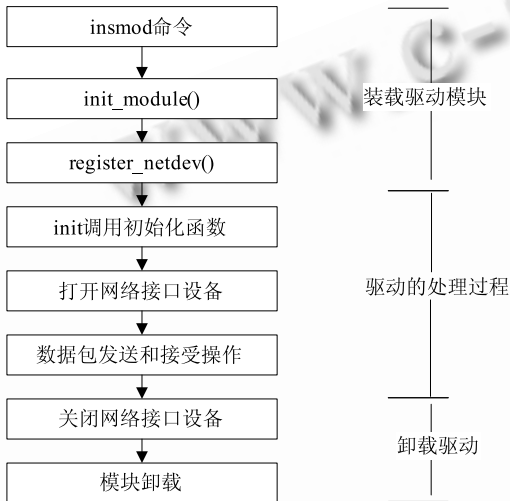


图 2 驱动的调度过程

5 总结

随着技术的日新月异,人们对于网络的需求愈发强烈,越来越多的嵌入式设备都需要具有以太网的接入功能,因此开发网络驱动程序对于很多嵌入式产品的研发至关重要。具体开发嵌入式 Linux 网络驱动程序时,可以参照内核中已经支持的网络驱动程序源代码,在重点理解 Linux 网络驱动实现原理的基础上,按照模块设计较为固定的开发模式,结合具体物理设备的硬件手册,移植编写需要的模块化的网络驱动程序。

本文在阐述嵌入式 Linux 网络通信模块驱动的基础上,结合实际的硬件平台优龙公司的 FS-PXA255 开发板,分析了 Linux2.6 内核下的 DM9000 网卡驱动程序,详细说明了嵌入式网络驱动程序的实现原理和开发流程,对在其它平台上开发嵌入式 Linux 网络驱动程序有一定的借鉴参考意义。

参考文献

- 1 DM9000-DS-F03 Data Sheet.Davicom Semiconductor Inc, 2004,7.
- 2 孙天泽,袁文菊,张海峰.嵌入式设计及 Linux 驱动开发指南:基于 ARM9 处理器.北京:电子工业出版社, 2005.
- 3 Davicom Corporation.DM9000 Application Notes V1. 20, 2005.