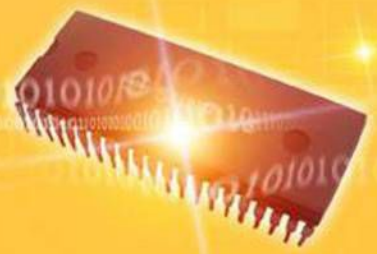
嵌入式系统工程师



Linux网卡驱动

大 纲

- Linux网络驱动概述
- linux网络驱动框架
- linux驱动数据结构及注册过程
- linux驱动数据包收发流程

大 纲

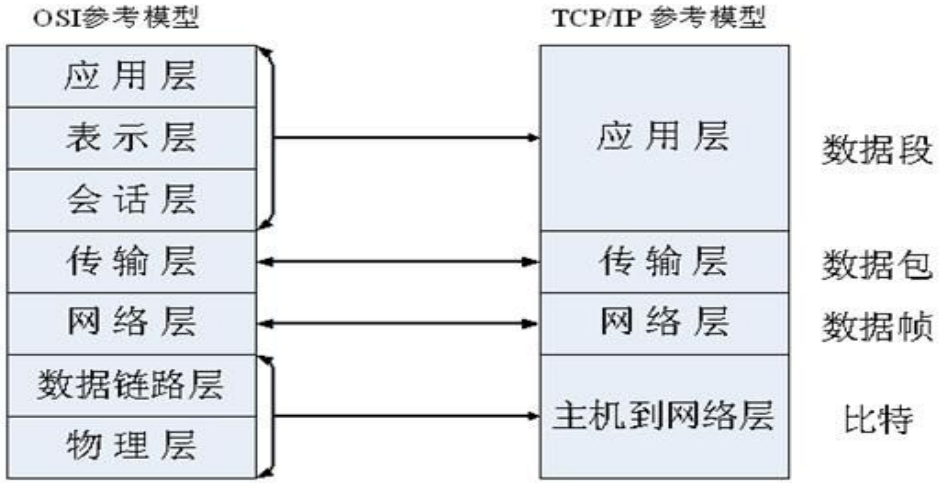
- Linux网络驱动概述
- linux网络驱动框架
- linux驱动数据结构及注册过程
- linux驱动数据包收发流程

Linux网络驱动概述

- 网络设备驱动是linux内核中三大类设备驱动之一，它用来完成高层网络协议的底层数据传输及设备控制
- 网络设备与其他两种设备的区别：
  - 网络接口不存在于/dev目录下，用户通过套接口socket函数使用网络
  - 网络除了响应来自内核的请求外，还需要处理外界的异步数据
  - 除了数据处理，网络设备还要完成地址设置、配置网络参数及流量统计等管理任务

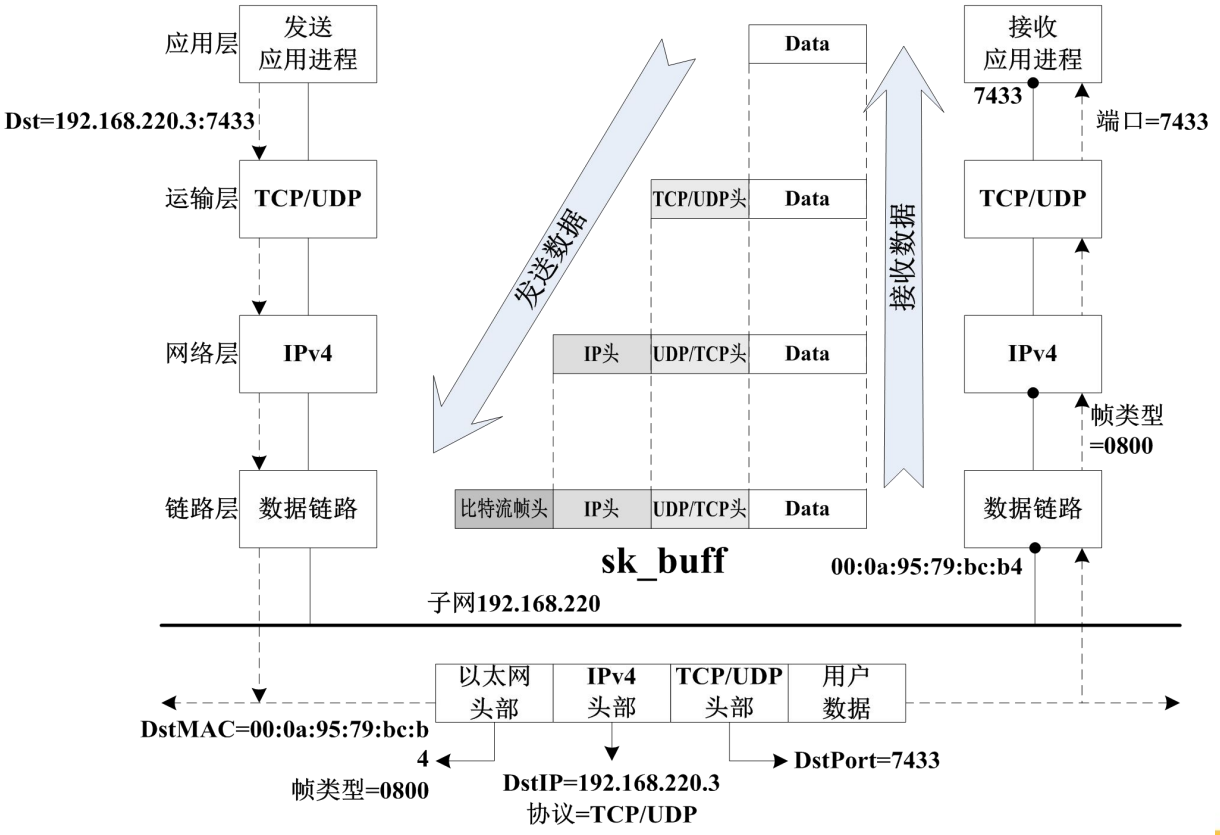
Linux网络驱动概述

- 提到网卡不得不提一下网络分层模型，在互联网发展过程中出现过两种协议分层模型，及OSI模型和TCP/IP参考模型，但事实上被采用的是TCP/IP模型
  - OSI模型和TCP/IP模型



Linux网络驱动概述

- 网络系统的数据传输使用了数据封装的方式，当网络数据包从上层往下层传输时，下层会加上本层协议头，继续传给下层。而数据包从下层向上层传输时，则过程相反

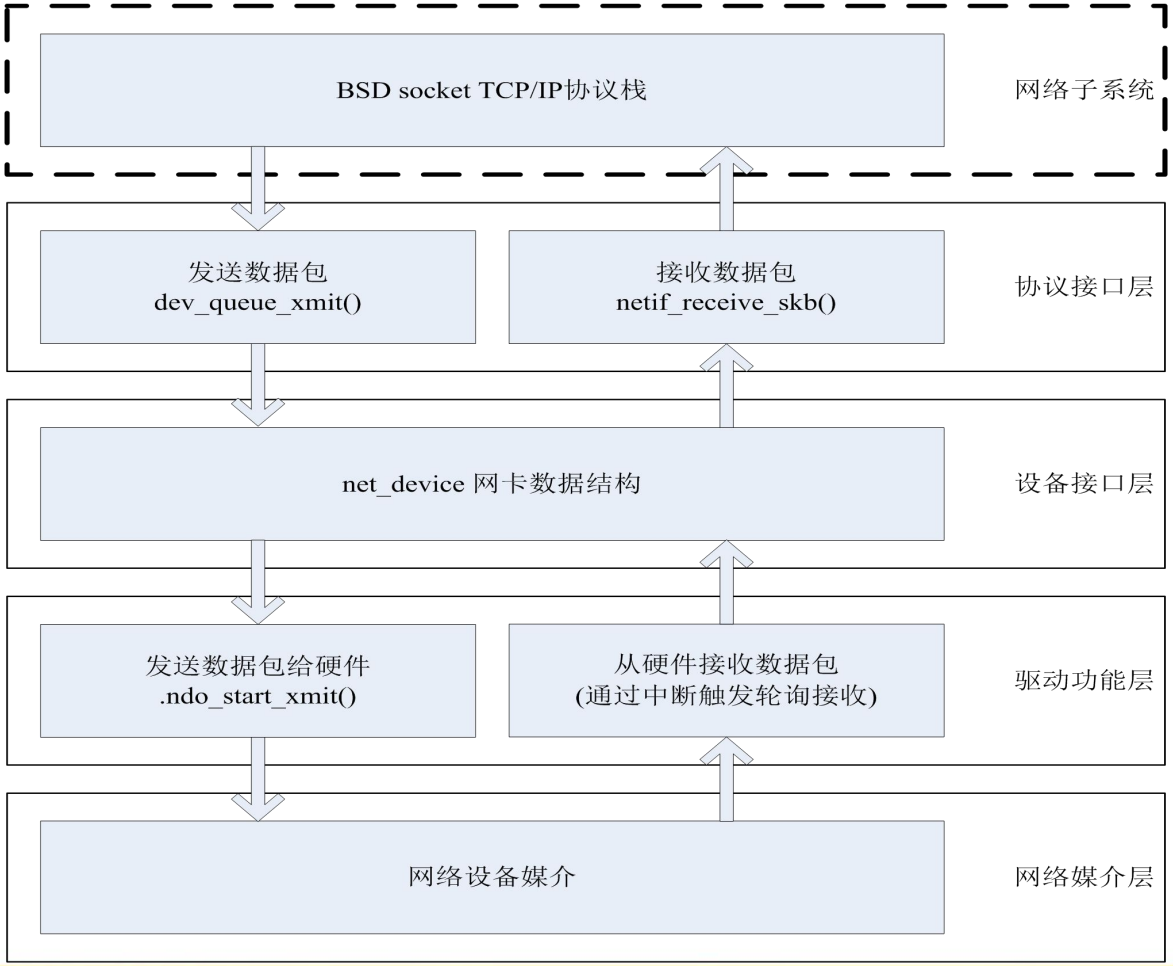


大 纲

- Linux网络驱动概述
- linux网络驱动框架
- linux驱动数据结构及注册过程
- linux驱动数据包收发流程

linux网络驱动框架

➢Linux网络设备驱动层次结构为：



linux网络驱动框架

- 网络协议接口层
  - 给上层协议提供统一的数据包收发接口，无论上层是ARP协议还是IP协议，都通过dev_queue_xmit函数发送数据，对数据包的接收通过netif_receive_skb函数实现，函数原型：
  - int dev_queue_xmit(struct sk_buff *skb)

    int netif_receive_skb(struct sk_buff *skb)
    - sk_buff是 套接字缓冲区，用于在linux网络子系统中各层间传输数据,网络子系统数据传递的“中枢神经”，发送数据时内核协议栈将建立好的sk_buff交给网络驱动部分，同样，当网络设备接收到数据，网络驱动要数据转换为sk_buff传给协议栈

linux网络驱动框架

- 网络设备接口层
  - 为千变万化的网络设备定义统一的、抽象的数据结构net_device结构体，实现多种硬件在软件层次上的统一，包含网络设备的属性描述和操作接口。
  - net_device结构体在内核中指代一个网络设备，网络设备驱动只需填充其结构体并注册到内核就可以实现内核与具体硬件操作函数的挂接。

linux网络驱动框架

- 设备驱动功能层
  - 对应net_device结构体中的设备驱动功能函数，如net_device_ops结构体包含的xxx_open()、xxx_stop()、xxx_tx()
- 网络设备与媒介层
  - 物理网卡通常包括PHY和MAC两个控制器，在OSI七层模型中，PHY指物理层，定义数据收发所需要的电气特性；MAC对应数据链路层，提供寻址机构、数据帧的构建、数据差错检查、传送控制

大 纲

- Linux网络驱动概述
- linux网络驱动框架
- linux驱动数据结构及注册过程
- linux驱动数据包收发流程

linux驱动数据结构

- 完成网络设备驱动本质上就是要填充net_device结构体，更确切说是填充部分结构体，其他大部分成员都是供内核使用。

- 例如比较重要的两个结构体：
  - net_device_ops 和ethtool_ops

linux驱动数据结构

```
static const struct net_device_ops smsc911x_netdev_ops = {
        .ndo_open               = smsc911x_open,
        .ndo_stop       = smsc911x_stop,
        .ndo_start_xmit             = smsc911x_hard_start_xmit,
        .ndo_get_stats          = smsc911x_get_stats,
        .ndo_set_multicast_list    = smsc911x_set_multicast_list,
        .ndo_do_ioctl           = smsc911x_do_ioctl,
        .ndo_change_mtu             = eth_change_mtu,
        .ndo_validate_addr  = eth_validate_addr,
        .ndo_set_mac_address    = smsc911x_set_mac_address,
#ifdef CONFIG_NET_POLL_CONTROLLER
        .ndo_poll_controller = smsc911x_poll_controller,
#endif
};
```

linux驱动数据结构

```
static const struct ethtool_ops smsc911x_ethtool_ops = {
        .get_settings = smsc911x_ethtool_getsettings,
        .set_settings = smsc911x_ethtool_setsettings,
        .get_link = ethtool_op_get_link,
        .get_drvinfo = smsc911x_ethtool_getdrvinfo,
        .nway_reset = smsc911x_ethtool_nwayreset,
        .get_msglevel = smsc911x_ethtool_getmsglevel,
        .set_msglevel = smsc911x_ethtool_setmsglevel,
        .get_regs_len = smsc911x_ethtool_getregslen,
        .get_regs = smsc911x_ethtool_getregs,
        .get_eeprom_len = smsc911x_ethtool_get_eeprom_len,
        .get_eeprom = smsc911x_ethtool_get_eeprom,
        .set_eeprom = smsc911x_ethtool_set_eeprom,
};
```

linux驱动数据结构

- 设备方法net_device_ops
  - 网络设备最核心的功能是收发数据包，此外还需要提供配置与统计功能，这些被称之为设备方法
  - 设备方法的实现依赖于具体的硬件环境，该部分正好对应了网卡驱动中的设备功能层
  - 我们从不同的硬件设备中提取具有共性的东西进行统一描述和操作，形成了net_device_ops设备方法

linux驱动注册过程

- 网络设备驱动的注册与注销
  - int register_netdev(struct net_device *dev)
    - 该函数完成注册网络设备驱动到linux内核，该函数会为设备分配一个接口名称，然后进行设备的注册
  - int unregister_netdev(struct net_device *dev)
    - 注销设备驱动，释放占用的系统资源

大 纲

- Linux网络驱动概述
- linux网络驱动框架
- linux驱动数据结构及注册过程
- linux驱动数据包收发流程

linux驱动数据包收发流程

- 数据收发流程概述：
  - 发送：发送数据时，linux内核的网络处理模块必须建立一个包含要传输的数据包的sk_buff，然后将sk_buff递交给下层，各层在sk_buff中添加不同的协议头直至交给网络设备发送。
  - 接收：当网络设备从媒介收到数据包后，它必须将数据转换为sk_buff数据结构并传递给上层，各层剥去相应的协议头直至交给用户。

linux驱动数据包收发流程

- 相关数据结构sk_buff
  - sk_buff称为“套接字缓冲区”，用于在Linux网络子系统中各层之间传递数据。是Linux网络子系统数据传递的“中枢神经”，是IP层与链路层交流的桥梁。
  - sk_buff的申请和释放
    - struct sk_buff *alloc_skb(unsigned int size, gfp_t priority)
    - struct sk_buff *dev_alloc_skb(unsigned int length)
    - void kfree_skb(struct sk_buff *skb)
    - void dev_kfree_skb(struct sk_buff *skb)

linux驱动数据包收发流程

➢ 数据包的发送流程

linux驱动数据包收发流程

➢ 数据包的接收流程

凌阳教育官方微信：Sunplusedu

Tel: 400-705-9680 , BBS: www.51develop.net , QQ群：241275518