

[首页](#) [资讯](#) [精华](#) [论坛](#) [问答](#) [博客](#) [专栏](#) [群组](#) [更多](#) ▼  
[您还未登录!](#) [登录](#) [注册](#)

## 指点江山

- [博客](#)
- [微博](#)
- [相册](#)
- [收藏](#)
- [留言](#)
- [关于我](#)

### netlink和rtnetlink (一)

**博客分类:**

- [Linux网络开发](#)

转到: <http://blogold.chinaunix.net/u/15993/showart.php?id=89359>

我们先从netlink说起，netlink其实就是一组宏，这组宏用来访问和创建netlink数据报，其实和其他套结字一样，只不过它是用来给用户进程和内核模块之间进行通信的，它的宏定义有：

```
#include <asm/types.h>
#include <linux/netlink.h>
int NLMSG_ALIGN(size_t len);
int NLMSG_LENGTH(size_t len);
int NLMSG_SPACE(size_t len);
void *NLMSG_DATA(struct nlmsghdr *nlh);
struct nlmsghdr *NLMSG_NEXT(struct nlmsghdr *nlh, int len);
int NLMSG_OK(struct nlmsghdr *nlh, int len);
int NLMSG_PAYLOAD(struct nlmsghdr *nlh, int len);
```

这些宏的含义我就不多说了，大家可以man 3 netlink。这里要说的是netlink的用法，netlink的用法在初始化的时候和一般的socket相似，如下：

```
#include <asm/types.h>
#include <sys/socket.h>
#include <linux/netlink.h>
netlink_socket = socket(PF_NETLINK, socket_type, netlink_family);
```

socket\_type为SOCK\_RAW或者SOCK\_DGRAM都可以，因为netlink本身是基于数据报的。  
netlink\_family有下面几种：

NETLINK\_ROUTE

用来修改和读取路由表的，这是我们后面要讨论的关键问题。

NETLINK\_FIREWALL

接收IPv4防火墙代码发送的信息。

NETLINK\_ARPD

在用户空间中管理ARP表。

NETLINK\_ROUTE6

接收和发送路由表更新。

**NETLINK\_IP6\_FW**

to receive packets that failed the IPv6 firewall checks (currently not implemented).

**NETLINK\_TAPBASE...NETLINK\_TAPBASE+15**

are the instances of the ethertap device. Ethertap is a pseudo network tunnel device that allows an ethernet driver to be simulated from user space.

**NETLINK\_SKIP**

Reserved for ENskip.

**NETLINK\_USERSOCK**

is reserved for future user space protocols.

Netlink消息是一个含有一个或者多个nlmsgghdr头和相关载荷组成的字节流。除了最后一个nlmsgghdr头的类型是NLMSG\_DONE，其他的nlmsgghdr头都是NLM\_F\_MULTI类型。这个字节流只能被上面描述过的标准NLMSG\_\*宏来访问。

Netlink是一个不可靠的协议。如果出现内存用尽或者其他错误，它就会丢包。如果需要可靠传输。发送者可以通过设置NLM\_F\_ACK标志 来向接收端要求一个ACK。一个ACK其实就是一个NLMSG\_ERROR包，只不过它的error field设置为0。应用程序需要为接收的消息自己产生ACK。内核尽力为每个失败的包产生NLMSG\_ERROR消息。用户进程也要遵守这个规范。

struct nlmsgghdr

```
{
    __u32  nlmsg_len; /* Length of message including header */
    __u16  nlmsg_type; /* Message content */
    __u16  nlmsg_flags; /* Additional flags */
    __u32  nlmsg_seq; /* Sequence number */
    __u32  nlmsg_pid; /* Sending process PID */
};
```

struct nlmsgerr

```
{
    int  error; /* negative errno or 0 for acks. */
    struct nlmsgghdr msg; /* message header that caused the error */
};
```

nlmsg\_flag的标准flag

NLM\_F\_REQUEST set on all request messages

NLM\_F\_MULTI the message is part of a multipart message terminated by NLMSG\_DONE

NLM\_F\_ACK reply with an acknowledgment on success

NLM\_F\_ECHO echo this request

GET请求的附加标志位

NLM\_F\_ROOT Return the complete table instead of a single entry.

NLM\_F\_MATCH Not implemented yet.

NLM\_F\_ATOMIC Return an atomic snapshot of the table.

NLM\_F\_DUMP not documented yet.

NEW请求的附加标志位

NLM\_F\_REPLACE Override existing object.

NLM\_F\_EXCL Don't replace if the object already exists.

NLM\_F\_CREATE Create object if it doesn't already exist.

NLM\_F\_APPEND Add to the end of the object list.

sockaddr\_nl结构描述了一个用户空间或者内核空间里面的netlink客户端。一个sockaddr\_nl可以为单播也可以为组播。



struct sockaddr\_nl

```
{
```

```
sa_family_t nl_family; /* AF_NETLINK */
unsigned short nl_pad; /* zero */
pid_t nl_pid; /* process pid */
__u32 nl_groups; /* multicast groups mask */
};
```

nl\_pid是用户空间netlink的pid, 如果为0, 就代表是内核空间。nl\_groups是一个位掩码, 每个bit代表了一个组号。如果是0就是单播。

( 未完待续 )

分享到:  

[netlink与rtnetlink \(二\)](#) | [Linux——Netlink](#)

- 2011-09-19 01:35
- 浏览 1480
- [评论\(0\)](#)
- 分类: [编程语言](#)
- [相关推荐](#)

评论



发表评论



[您还没有登录,请您登录后再发表评论](#)



haohetao

- 浏览: 393200 次
- 性别: 
- 来自: 郑州
-  我现在离线

最近访客

[更多访客>>](#)



[dylinshi126](#)



[dongbiying](#)



[wwwgui](#)

[sunburg](#)

## 文章分类

- [全部博客 \(474\)](#)
- [数据库 \(25\)](#)
- [操作系统 \(127\)](#)
- [PHP \(12\)](#)
- [前端 \(13\)](#)
- [手机 \(6\)](#)
- [电脑应用 \(30\)](#)
- [Linux网络开发 \(27\)](#)
- [C/C++ \(15\)](#)

## 社区版块

- [我的资讯](#) (0)
- [我的论坛](#) (6)
- [我的问答](#) (0)

## 存档分类

- [2014-08](#) (1)
- [2014-03](#) (3)
- [2014-01](#) (1)
- [更多存档...](#)

## 最新评论

- [exception01](#): 不错, 可以基本了解 概念 +1  
[什么是动态语言和静态语言?](#)
- [nature\\_XD](#): 作者前两句解释的很清晰, 赞!  
[SMI接口, SMI帧结构, MDC/MDIO](#)
- [chumignze](#): 楼主能给我发个实例吗, 现在急需解决这个问题, 谢谢3071081 ...  
[highcharts.highstock用ajax延迟动态加载数据](#)
- [yaguang\\_li](#): 果然是有道划词搜索, 谢谢。  
[SecureCRT鼠标双击或拖成变成Ctrl+C的解决方法](#)
- [pennyxi](#): ...  
[DreamWeaver CS5 中文版可激活注册码](#)

---

声明: ITeye文章版权属于作者, 受法律保护。没有作者书面许可不得转载。若作者同意转载, 必须以超链接形式标明文章原始出处和作者。

© 2003-2014 ITeye.com. All rights reserved. [ 京ICP证110151号 京公网安备110105010620 ]