## NAME

libnetlink − A library for accessing the netlink service

## SYNOPSIS

```
#include <asm/types.h>
#include <libnetlink.h>
#include <linux/netlink.h>
#include <linux/rtnetlink.h>

int rtnl_open(struct rtnl_handle *rth, unsigned subscriptions)

int rtnl_wilddump_request(struct rtnl_handle *rth, int family, int type)

int rtnl_send(struct rtnl_handle *rth, char *buf, int len)

int rtnl_dump_request(struct rtnl_handle *rth, int type, void *req, int len)

int rtnl_dump_filter(struct rtnl_handle *rth,
                     int (*filter)(struct sockaddr_nl *, struct nlmsghdr *n, void *),
                     void *arg1,
                     int (*junk)(struct sockaddr_nl *,struct nlmsghdr *n, void *),
                     void *arg2)

int rtnl_talk(struct rtnl_handle *rtnl, struct nlmsghdr *n, pid_t peer,
              unsigned groups, struct nlmsghdr *answer,
              int (*junk)(struct sockaddr_nl *,struct nlmsghdr *n, void *),
              void *jarg)

int rtnl_listen(struct rtnl_handle *rtnl,
              int (*handler)(struct sockaddr_nl *, struct rtnl_ctrl_data *,
                             struct nlmsghdr *n, void *),
              void *jarg)

int rtnl_from_file(FILE *rtnl,
              int (*handler)(struct sockaddr_nl *,struct nlmsghdr *n, void *),
              void *jarg)

int addattr32(struct nlmsghdr *n, int maxlen, int type, __u32 data)

int addattr_l(struct nlmsghdr *n, int maxlen, int type, void *data, int alen)

int rta_addattr32(struct rtattr *rta, int maxlen, int type, __u32 data)

int rta_addattr_l(struct rtattr *rta, int maxlen, int type, void *data, int alen)
```

## DESCRIPTION

libnetlink provides a higher level interface to **rtnetlink(7).** The read functions return 0 on success and a negative errno on failure. The send functions return the amount of data sent, or -1 on error.

rtnl_open

Open a rtnetlink socket and save the state into the **rth** handle. This handle is passed to all subsequent calls. **subscriptions** is a bitmap of the rtnetlink multicast groups the socket will be a member of.

rtnl_wilddump_request
: Request a full dump of the **type** database for **family** addresses. **type** is a rtnetlink message type.

rtnl_dump_request
: Request a full dump of the **type** data buffer into **buf** with maximum length of **len. type** is a rtnetlink message type.

rtnl_dump_filter
: Receive netlink data after a request and filter it. The **filter** callback checks if the received message is wanted. It gets the source address of the message, the message itself and **arg1** as arguments. 0 as return means that the filter passed, a negative value is returned by *rtnl_dump_filter* in case of error. NULL for *filter* means to not use a filter. **junk** is used to filter messages not destined to the local socket. Only one message bundle is received. If there is a message pending, this function does not block.

rtnl_listen
: Receive netlink data after a request and pass it to *handler.* **handler** is a callback that gets the message source address, anscillary data, the message itself, and the **jarg** cookie as arguments. It will get called for all received messages. Only one message bundle is received. If there is a message pending this function does not block.

rtnl_from_file
: Works like *rtnl_listen,* but reads a netlink message bundle from the file **file** and passes the messages to **handler** for parsing. The file should contain raw data as received from a rtnetlink socket.

The following functions are useful to construct custom rtnetlink messages. For simple database dumping with filtering it is better to use the higher level functions above. See **rtnetlink(3)** and **netlink(3)** on how to generate a rtnetlink message. The following utility functions require a continuous buffer that already contains a netlink message header and a rtnetlink request.

rtnl_send
: Send the rtnetlink message in **buf** of length **len** to handle **rth.**

addattr32
: Add a __u32 attribute of type **type** and with value **data** to netlink message **n,** which is part of a buffer of length **maxlen.**

addattr_l
: Add a variable length attribute of type **type** and with value **data** and **alen** length to netlink message **n,** which is part of a buffer of length **maxlen. data** is copied.

rta_addattr32
: Initialize the rtnetlink attribute **rta** with a __u32 data value.

rta_addattr32
: Initialize the rtnetlink attribute **rta** with a variable length data value.

## BUGS

This library is meant for internal use, use libmnl for new programs.

The functions sometimes use fprintf and exit when a fatal error occurs. This library should be named

librtnetlink.

## AUTHORS

netlink/rtnetlink was designed and written by Alexey Kuznetsov. Andi Kleen wrote the man page.

## SEE ALSO

**netlink(7), rtnetlink(7)**
/usr/include/linux/rtnetlink.h