**NAME**
       netlink − Netlink macros

**SYNOPSIS**
       **#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***);**

**DESCRIPTION**
       *<linux/netlink.h>* defines several standard macros to access or create a netlink datagram. They are similar
       in spirit to the macros defined in cmsg(3) for auxiliary data. The buffer passed to and from a netlink socket
       should be accessed using only these macros.

       **NLMSG_ALIGN**()
              Round the length of a netlink message up to align it properly.

       **NLMSG_LENGTH**()
              Given the payload length, *len*, this macro returns the aligned length to store in the *nlmsg_len* field
              of the *nlmsghdr*.

       **NLMSG_SPACE**()
              Return the number of bytes that a netlink message with payload of *len* would occupy.

       **NLMSG_DATA**()
              Return a pointer to the payload associated with the passed *nlmsghdr*.

       **NLMSG_NEXT**()
              Get the next *nlmsghdr* in a multipart message. The caller must check if the current *nlmsghdr*
              didn't have the **NLMSG_DONE** set—this function doesn't return NULL on end. The *len* argu-
              ment is an lvalue containing the remaining length of the message buffer. This macro decrements it
              by the length of the message header.

       **NLMSG_OK**()
              Return true if the netlink message is not truncated and is in a form suitable for parsing.

       **NLMSG_PAYLOAD**()
              Return the length of the payload associated with the *nlmsghdr*.

**CONFORMING TO**
       These macros are nonstandard Linux extensions.

**NOTES**
       It is often better to use netlink via *libnetlink* than via the low-level kernel interface.

**SEE ALSO**
       libnetlink(3), netlink(7)

**COLOPHON**
       This page is part of release 4.16 of the Linux *man-pages* project. A description of the project, information
       about   reporting   bugs,   and   the   latest   version   of   this   page,   can   be   found   at
       https://www.kernel.org/doc/man−pages/.