NAME

insq – insert a message into a queue

SYNOPSIS

#include <sys/stream.h>

int insq(queue_t **q*, **mblk_t** **emp*, **mblk_t** **nmp***);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

q Pointer to the queue containing message *emp*.

emp Enqueued message before which the new message is to be inserted. **mblk_t** is an instance of the **msgb**(9S) structure.

nmp Message to be inserted.

DESCRIPTION

insq( ) inserts a message into a queue. The message to be inserted, *nmp*, is placed in *q* immediately before the message *emp*. If *emp* is **NULL**, the new message is placed at the end of the queue. The queue class of the new message is ignored. All flow control parameters are updated. The service procedure is enabled unless **QNOENB** is set.

RETURN VALUES

insq( ) returns **1** on success, and **0** on failure.

CONTEXT

insq( ) can be called from user or interrupt context.

EXAMPLE

This routine illustrates the steps a transport provider may take to place expedited data ahead of normal data on a queue (assume all **M_DATA** messages are converted into **M_PROTO T_DATA_REQ** messages). Normal **T_DATA_REQ** messages are just placed on the end of the queue (line 16). However, expedited **T_EXDATA_REQ** messages are inserted before any normal messages already on the queue (line 25). If there are no normal messages on the queue, **bp** will be **NULL** and we fall out of the **for** loop (line 21). **insq** acts like **putq**(9F) in this case.

```
 1 #include <sys/tihdr.h>
 2 #include <sys/stream.h>
 3
 4 static int
 5 xxxwput(queue_t *q, mblk_t *mp)
 6 {
 7         union T_primitives *tp;
 8         mblk_t *bp;
 9         union T_primitives *ntp;
10
11         switch (mp->b_datap->db_type) {
12         case M_PROTO:
13                 tp = (union T_primitives *)mp->b_rptr;
14                 switch (tp->type) {
15                 case T_DATA_REQ:
16                         putq(q, mp);
```

```
                        break;

                case T_EXDATA_REQ:
                        freezestr(q);
                        for (bp = q->q_first; bp; bp = bp->b_next) {
                          if (bp->b_datap->db_type == M_PROTO) {
                            ntp = (union T_primitives *)bp->b_rptr;
                            if (ntp->type != T_EXDATA_REQ)
                              break;
                          }
                        }
                        (void) insq(q, bp, mp);
                        unfreezestr(q);
                        break;
                . . .
        }
    }
}
```

SEE ALSO

freezestr(9F), **putq**(9F), **rmvq**(9F), **unfreezestr**(9F), **msgb**(9S)

Writing Device Drivers
STREAMS Programming Guide

WARNINGS

If *emp* is non-NULL, it must point to a message on *q* or a system panic could result.

NOTES

The stream must be frozen using **freezestr**(9F) before calling **insq( )**.