NAME

qreply – send a message on a stream in the reverse direction

SYNOPSIS

```
#include <sys/stream.h>
void qreply(queue_t *q, mblk_t *mp);
```

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

q Pointer to the queue.

mp Pointer to the message to be sent in the opposite direction.

DESCRIPTION

qreply( ) sends messages in the reverse direction of normal flow. That is, **qreply(***q***,** *mp***)** is equivalent to **putnext(OTHERQ(***q***),** *mp***).**

CONTEXT

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

EXAMPLE

This example depicts the canonical flushing code for STREAMS drivers. Assume that the driver has service procedures (see **srv**(9E)), so that there may be messages on its queues. Its write-side put procedure (see **put**(9E)) handles **M_FLUSH** messages by first checking the **FLUSHW** bit in the first byte of the message, then the write queue is flushed (line 8) and the **FLUSHW** bit is turned off (line 9). If the **FLUSHR** bit is on, then the read queue is flushed (line 12) and the message is sent back up the read side of the stream with the **qreply**(9F) function (line 13). If the **FLUSHR** bit is off, then the message is freed (line 15). See the example for **flushq**(9F) for the canonical flushing code for modules.

```
 1 xxxwput(q, mp)
 2    queue_t *q;
 3    mblk_t *mp;
 4 {
 5        switch(mp->b_datap->db_type) {
 6        case M_FLUSH:
 7                if (*mp->b_rptr & FLUSHW) {
 8                        flushq(q, FLUSHALL);
 9                        *mp->b_rptr &= ~FLUSHW;
10                }
11                if (*mp->b_rptr & FLUSHR) {
12                        flushq(RD(q), FLUSHALL);
13                        qreply(q, mp);
14                } else {
15                        freemsg(mp);
16                }
17                break;
          . . .
18        }
19 }
```

SEE ALSO

put(9E), **srv**(9E), **flushq**(9F), **OTHERQ**(9F), **putnext**(9F)

Writing Device Drivers
STREAMS Programming Guide