NAME

put – receive messages from the preceding queue

SYNOPSIS

#include <sys/types.h>
#include <sys/stream.h>
#include <sys/stropts.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>

int *prefix***rput(queue_t** **q***, mblk_t** **mp***); /* read side */**

int *prefix***wput(queue_t** **q***, mblk_t** **mp***); /* write side */**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI). This entry point is **required** for STREAMS.

ARGUMENTS

q Pointer to the **queue**(9S) structure.

mp Pointer to the message block.

DESCRIPTION

The primary task of the **put()** routine is to coordinate the passing of messages from one queue to the next in a stream. The **put()** routine is called by the preceding stream component (stream module, driver, or stream head). **put()** routines are designated ''write'' or ''read'' depending on the direction of message flow.

With few exceptions, a streams module or driver must have a **put()** routine. One exception is the read side of a driver, which does not need a **put()** routine because there is no component downstream to call it. The **put()** routine is always called before the component's corresponding **srv**(9E) (service) routine, and so **put()** should be used for the immediate processing of messages.

A **put()** routine must do at least one of the following when it receives a message:

- pass the message to the next component on the stream by calling the **putnext**(9F) function
- process the message, if immediate processing is required (for example, to handle high priority messages)
- enqueue the message (with the **putq**(9F) function) for deferred processing by the service **srv**(9E) routine

Typically, a **put()** routine will switch on message type, which is contained in the **db_type** member of the **datab** structure pointed to by *mp*. The action taken by the **put()** routine depends on the message type. For example, a **put()** routine might process high priority messages, enqueue normal messages, and handle an unrecognized **M_IOCTL** message by changing its type to **M_IOCNAK** (negative acknowledgement) and sending it back to the stream head using the **qreply**(9F) function.

The **putq**(9F) function can be used as a module's **put( )** routine when no special processing is required and all messages are to be enqueued for the **srv** (9E) routine.

RETURN VALUES Ignored.

CONTEXT **put( )** routines do not have user context.

SEE ALSO **srv**(9E), **putctl**(9F), **putctl1**(9F), **putnext**(9F), **putnextctl**(9F), **putnextctl1**(9F), **putq**(9F), **qreply**(9F), **streamtab**(9S)

Writing Device Drivers
STREAMS Programming Guide