NAME

testb – check for an available buffer

SYNOPSIS

#include <sys/stream.h>

int testb(int *size***, unsigned int** *pri***);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

size	Size of the requested buffer.
pri	Priority of the allocb request.

DESCRIPTION

testb( ) checks to see if an **allocb**(9F) call is likely to succeed if a buffer of *size* bytes at priority *pri* is requested. Even if **testb( )** returns successfully, the call to **allocb** (9F) can fail. The *pri* argument is no longer used, but is retained for compatibility.

RETURN VALUES

Returns **1** if a buffer of the requested size is available, and **0** if one is not.

CONTEXT

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

EXAMPLES

In a service routine, if **copymsg**(9F) fails (line 6), the message is put back on the queue (line 7) and a routine, **tryagain**, is scheduled to be run in one tenth of a second. Then the service routine returns.

When the **timeout**(9F) function runs, if there is no message on the front of the queue, it just returns. Otherwise, for each message block in the first message, check to see if an allocation would succeed. If the number of message blocks equals the number we can allocate, then enable the service procedure. Otherwise, reschedule **tryagain** to run again in another tenth of a second. Note that **tryagain** is merely an approximation. Its accounting may be faulty. Consider the case of a message comprised of two 1024-byte message blocks. If there is only one free 1024-byte message block and no free 2048-byte message blocks, then **testb( )** will still succeed twice. If no message blocks are freed of these sizes before the service procedure runs again, then the **copymsg**(9F) will still fail. The reason **testb( )** is used here is because it is significantly faster than calling **copymsg**. We must minimize the amount of time spent in a **timeout** routine.

```
 1  xxxsrv(q)
 2     queue_t *q;
 3  {
 4        mblk_t *mp;
 5        mblk_t *nmp;
      . . .
 6        if ((nmp = copymsg(mp)) == NULL) {
 7                  putbq(q, mp);
 8                  timeout(tryagain, (long)q, drv_usectohz(100000));
 9                  return;
10        }
          . . .
```

```
}

tryagain(q)
   queue_t *q;
{
       register int can_alloc = 0;
       register int num_blks = 0;
       register mblk_t *mp;

       if (!q->q_first)
               return;
       for (mp = q->q_first; mp; mp = mp->b_cont) {
               num_blks++;
               can_alloc += testb((mp->b_datap->db_lim -
                  mp->b_datap->db_base), BPRI_MED);
       }
       if (num_blks == can_alloc)
               qenable(q);
       else
               timeout(tryagain, (long)q, drv_usectohz(100000));
}
```

SEE ALSO **allocb**(9F), **bufcall**(9F), **copymsg**(9F)

Writing Device Drivers
STREAMS Programming Guide

NOTES The *pri* argument is provided for compatibility only. Its value is ignored.