NAME

bcopy – copy data between address locations in the kernel

SYNOPSIS

#include <sys/types.h>

void bcopy(caddr_t *from*, **caddr_t** *to*, **size_t** *bcount***);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

from Source address from which the copy is made.

to Destination address to which copy is made.

bcount The number of bytes moved.

DESCRIPTION

bcopy( ) copies *bcount* bytes from one kernel address to another. If the input and output addresses overlap, the command executes, but the results may not be as expected.

Note that **bcopy( )** should never be used to move data in or out of a user buffer, because it has no provision for handling page faults. The user address space can be swapped out at any time, and **bcopy( )** always assumes that there will be no paging faults. If **bcopy( )** attempts to access the user buffer when it is swapped out, the system will panic. It is safe to use **bcopy( )** to move data within kernel space, since kernel space is never swapped out.

CONTEXT

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

EXAMPLES

An I/O request is made for data stored in a RAM disk. If the I/O operation is a read request, the data is copied from the RAM disk to a buffer (line 8). If it is a write request, the data is copied from a buffer to the RAM disk (line 15). **bcopy( )** is used since both the RAM disk and the buffer are part of the kernel address space.

```
1  #define RAMDNBLK       1000            /* blocks in the RAM disk */
2  #define RAMDBSIZ       512             /* bytes per block */
3  char ramdblks[RAMDNBLK][RAMDBSIZ];     /* blocks forming RAM */
                                          /* disk */
     ...
4
5  if  (bp->b_flags & B_READ)             /* if read request, copy data */
6                                         /* from RAM disk data block */
7                                         /* to system buffer */
8          bcopy(&ramdblks[bp->b_blkno][0], bp->b_un.b_addr,
9                  bp->b_bcount);
10
11  else                                  /* else write request, */
12                                        /* copy data from a */
13                                        /* system buffer to RAM disk */
14                                        /* data block */
```

```
        bcopy(bp->b_un.b_addr, &ramdblks[bp->b_blkno][0],
            bp->b_bcount);
```

WARNINGS

The *from* and *to* addresses must be within the kernel space. No range checking is done. If an address outside of the kernel space is selected, the driver may corrupt the system in an unpredictable way.

SEE ALSO

copyin(9F), **copyout**(9F)

Writing Device Drivers