```
/* freebuf.c - freebuf */

#include <xinu.h>

/*------------------------------------------------------------------------
 *  freebuf  -  Free a buffer that was allocated from a pool by getbuf
 *------------------------------------------------------------------------
 */
syscall freebuf(
          char          *bufaddr        /* Address of buffer to return  */
        )
{
        intmask mask;                   /* Saved interrupt mask         */
        struct  bpentry *bpptr;         /* Pointer to entry in buftab   */
        bpid32  poolid;                 /* ID of buffer's pool          */

        mask = disable();

        /* Extract pool ID from integer prior to buffer address */

        bufaddr -= sizeof(bpid32);
        poolid = *(bpid32 *)bufaddr;
        if (poolid < 0  ||  poolid >= nbpools) {
                restore(mask);
                return SYSERR;
        }

        /* Get address of correct pool entry in table */

        bpptr = &buftab[poolid];

        /* Insert buffer into list and signal semaphore */

        ((struct bpentry *)bufaddr)->bpnext = bpptr->bpnext;
        bpptr->bpnext = (struct bpentry *)bufaddr;
        signal(bpptr->bpsem);
        restore(mask);
        return OK;
}
```

Recall that when it allocates a buffer, *getbuf* stores the pool ID in the four bytes that precede the buffer address. *Freebuf* moves back four bytes from the beginning of the buffer, and extracts the pool ID. After verifying that the pool ID is valid, *freebuf* uses the ID to locate the entry in the table of buffer pools. It then links the buffer back