

```

    lfptr->lfddirty = FALSE;

    signal(Lf_data.lf_mutex);

    return lfptr->lfdev;
}

```

After verifying that the length of the file name is valid, *lfsopen* calls *lfgetmode* to parse the *mode* argument and convert it to a set of bits. A mode argument consists of a null-terminated string that contains zero or more of the characters from Figure 19.5. Characters in the mode string may not be repeated, and the combination of “o” and “n” (i.e., file is both *old* and *new*) is considered illegal. Furthermore, if neither “r” nor “w” is present in the mode string, *lfgetmode* assumes a default mode that allows both reading and writing. After parsing the mode string, *lfgetmode* produces an integer with individual bits specifying the modes as defined in *lfilesys.h*.[†] File *lfgetmode.c* contains the code.

Character	Meaning
r	Open the file for reading
w	Open the file for writing
o	File must be “old” (i.e., must exist)
n	File must be “new” (i.e., must not exist)

Figure 19.5 Characters permitted in a mode string and their meaning.

```

/* lfgetmode.c - lfgetmode */

#include <xinu.h>

/*-----
 * lfgetmode - Parse mode argument and generate integer of mode bits
 *-----
 */
int32 lfgetmode (
    char *mode          /* String of mode characters */
)
{
    int32 mbits;        /* Mode bits to return */
    char ch;           /* Next char in mode string */

```

[†]File *lfilesys.h* can be found on page 464.