

LCOV - code coverage report

Current view: [directory](#) - [fs/ext4](#) - [ext4_jbd2.c](#) ([source](#) / [functions](#))

Found Hit

Test: [kernel_2_6_31_ext4_round_3.info](#)

Lines: 48 30

Date: 2009-10-24

Functions: 6 5

```
1      : /*
2      :  * Interface between ext4 and JBD
3      :  */
4      :
5      : #include "ext4_jbd2.h"
6      :
7      : int __ext4_journal_get_undo_access(const char *where, handle_t *handle,
8      :                                struct buffer_head *bh)
9      0 : {
10     0 :     int err = 0;
11     :
12     0 :     if (ext4_handle_valid(handle)) {
13     0 :         err = jbd2_journal_get_undo_access(handle, bh);
14     0 :         if (err)
15     0 :             ext4_journal_abort_handle(where, __func__, bh,
16     :                                     handle, err);
17     :     }
18     0 :     return err;
19     : }
20     :
21     : int __ext4_journal_get_write_access(const char *where, handle_t *handle,
22     :                                  struct buffer_head *bh)
23     2103165255 : {
24     2103165255 :     int err = 0;
25     :
26     2103165255 :     if (ext4_handle_valid(handle)) {
27     2103610168 :         err = jbd2_journal_get_write_access(handle, bh);
28     2127170016 :         if (err)
29     0 :             ext4_journal_abort_handle(where, __func__, bh,
30     :                                     handle, err);
31     :     }
32     2126725103 :     return err;
33     : }
34     :
35     : int __ext4_journal_forget(const char *where, handle_t *handle,
36     :                          struct buffer_head *bh)
37     179 : {
38     179 :     int err = 0;
39     :
40     179 :     if (ext4_handle_valid(handle)) {
41     179 :         err = jbd2_journal_forget(handle, bh);
42     179 :         if (err)
43     0 :             ext4_journal_abort_handle(where, __func__, bh,
44     :                                     handle, err);
45     :     }
46     :     else
47     :         brelse(bh);
48     179 :     return err;
49     : }
50     :
51     : int __ext4_journal_revoke(const char *where, handle_t *handle,
52     :                          ext4_fsblk_t blocknr, struct buffer_head *bh)
```

```

53      877105 : {
54      877105 :         int err = 0;
55      :
56      877105 :         if (ext4_handle_valid(handle)) {
57      877105 :             err = jbd2_journal_revoke(handle, blocknr, bh);
58      877105 :             if (err)
59      0 :                 ext4_journal_abort_handle(where, __func__, bh,
60      :                                     handle, err);
61      :         }
62      :         else
63      :             brelse(bh);
64      877105 :         return err;
65      :     }
66      :
67      : int __ext4_journal_get_create_access(const char *where,
68      :                                     handle_t *handle, struct buffer_head *bh)
69      1887214 : {
70      1887214 :         int err = 0;
71      :
72      1887214 :         if (ext4_handle_valid(handle)) {
73      1887342 :             err = jbd2_journal_get_create_access(handle, bh);
74      1888609 :             if (err)
75      0 :                 ext4_journal_abort_handle(where, __func__, bh,
76      :                                     handle, err);
77      :         }
78      1888481 :         return err;
79      :     }
80      :
81      : int __ext4_handle_dirty_metadata(const char *where, handle_t *handle,
82      :                                     struct inode *inode, struct buffer_head *bh)
83      2104103745 : {
84      2104103745 :         int err = 0;
85      :
86      2104103745 :         if (ext4_handle_valid(handle)) {
87      2104103745 :             err = jbd2_journal_dirty_metadata(handle, bh);
88      2127944433 :             if (err)
89      0 :                 ext4_journal_abort_handle(where, __func__, bh,
90      :                                     handle, err);
91      :         } else {
92      0 :             mark_buffer_dirty(bh);
93      0 :             if (inode && inode_needs_sync(inode)) {
94      0 :                 sync_dirty_buffer(bh);
95      0 :                 if (buffer_req(bh) && !buffer_uptodate(bh)) {
96      0 :                     ext4_error(inode->i_sb, __func__,
97      :                             "IO error syncing inode, "
98      :                             "inode=%lu, block=%llu",
99      :                             inode->i_ino,
100      :                             (unsigned long long) bh->b_blocknr);
101      0 :                     err = -EIO;
102      :                 }
103      :             }
104      :         }
105      2127944433 :         return err;
106      :     }

```