
Subject: [PATCH] reiserfs: add missing D-cache flushing
Posted by [Alexey Dobriyan](#) on Wed, 15 Nov 2006 07:42:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Looks like, reiserfs_prepare_file_region_for_write() doesn't contain several flush_dcache_page() calls.

Found with help from Dmitriy Monakhov <dmonakhov@openvz.org>

Dmitriy adds:

```
> Absolutely agree. This function is analog of generic __block_prepare_write()
>
> comment before this block may confuse us:
> ---reiserfs/file.c:1039
> /* Now if all the write area lies past the file end, no point in
>    mapping blocks, since there is none, so we just zero out remaining
>    parts of first and last pages in write area (if needed) */
>    if ((pos & ~((loff_t) PAGE_CACHE_SIZE - 1)) > inode->i_size) {
> ---
> but we can create mapping before file size was reduced and grow again
```

Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>

fs/reiserfs/file.c | 4 ++++
1 file changed, 4 insertions(+)

```
--- a/fs/reiserfs/file.c
+++ b/fs/reiserfs/file.c
@@ -1043,6 +1043,7 @@ static int reiserfs_prepare_file_region_
    if (from != 0) { /* First page needs to be partially zeroed */
        char *kaddr = kmap_atomic(prepared_pages[0], KM_USER0);
        memset(kaddr, 0, from);
+   flush_dcache_page(prepared_pages[0]);
        kunmap_atomic(kaddr, KM_USER0);
    }
    if (to != PAGE_CACHE_SIZE) { /* Last page needs to be partially zeroed */
@@ -1050,6 +1051,7 @@ static int reiserfs_prepare_file_region_
        kmap_atomic(prepared_pages[num_pages - 1],
            KM_USER0);
        memset(kaddr + to, 0, PAGE_CACHE_SIZE - to);
+   flush_dcache_page(prepared_pages[num_pages - 1]);
        kunmap_atomic(kaddr, KM_USER0);
    }

@@ -1183,6 +1185,7 @@ static int reiserfs_prepare_file_region_
    KM_USER0);
    memset(kaddr + block_start, 0,
```

```

        from - block_start);
+   flush_dcache_page(prepared_pages[0]);
    kunmap_atomic(kaddr, KM_USER0);
    set_buffer_uptodate(bh);
}
@@ -1220,6 +1223,7 @@ static int reiserfs_prepare_file_region_
    [num_pages - 1],
    KM_USER0);
    memset(kaddr + to, 0, block_end - to);
+   flush_dcache_page(prepared_pages[num_pages - 1]);
    kunmap_atomic(kaddr, KM_USER0);
    set_buffer_uptodate(bh);
}

```
