

Fix unlock_buffer() to work the same way as bit_unlock()

Fix unlock_buffer() to work the same way as bit_unlock()

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-03/msg09462.html>

- *From:* Christoph Lameter <clameter@xxxxxxx>
 - *Date:* Mon, 27 Mar 2006 19:59:02 -0800 (PST)
-

Currently unlock_buffer() contains a smb_mb__after_clear_bit() which is weird because bit_spin_unlock() uses smb_mb__before_clear_bit():

From include/linux/bit_spinlock.h:

```
static inline void bit_spin_unlock(int bitnum, unsigned long *addr)
{
    smb_mb__before_clear_bit();
    clear_bit(bitnum, addr);
    preempt_enable();
    __release(bitlock);
}
```

For most architectures there is no difference because both smb_mb__after_clear_bit() and smb_mb__before_clear_bit() are both memory barriers and clear_buffer_locked() is an atomic operation. However, they differ under IA64.

Note that this potential race has never been seen under IA64. It was discovered by inspection by Zoltan Menyhart <Zoltan.Menyhart@xxxxxxx>.

Regardless if this is a true race or not, I think the unlock sequence needs to be the same for bit locks and unlock_buffer(). Maybe unlock_buffer and lock_buffer better use bit spinlock operations?

Change unlock_buffer() to work the same way as bit_spin_unlock.

Signed-off-by: Christoph Lameter <clameter@xxxxxxx>

Index: linux-2.6/fs/buffer.c

```
=====
--- linux-2.6.orig/fs/buffer.c 2006-03-27 14:09:54.000000000 -0800
+++ linux-2.6/fs/buffer.c 2006-03-27 19:40:32.000000000 -0800
@@ -78,8 +78,8 @@ EXPORT_SYMBOL(__lock_buffer);
```

```
void fastcall unlock_buffer(struct buffer_head *bh)
```

Fix unlock_buffer() to work the same way as bit_unlock()

Fix unlock_buffer() to work the same way as bit_unlock()

```
{  
+ smp_mb__before_clear_bit();  
clear_buffer_locked(bh);  
- smp_mb__after_clear_bit();  
wake_up_bit(&bh->b_state, BH_Lock);  
}
```

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>