

> On Wed, 2012-10-17 at 17:18 +0400, Stanislav Kinsbursky wrote:

>

```
>> +static int posix_timer_add(struct k_itimer *timer)
```

```
>> +{
```

```
>> + struct signal_struct *sig = current->signal;
```

```
>> + int next_free_id = sig->posix_timer_id;
```

```
>> + struct hlist_head *head;
```

```
>> + int ret = -ENOENT;
```

```
>> +
```

```
>> + do {
```

```
>> + spin_lock(&hash_lock);
```

```
>> + head = &posix_timers_hashtable[hash(sig, sig->posix_timer_id)];
```

```
>> + if (__posix_timers_find(head, sig, sig->posix_timer_id) == NULL) {
```

```
>> + hlist_add_head_rcu(&timer->t_hash, head);
```

```
>
```

> Hmm...

>

```
>> + ret = sig->posix_timer_id++;
```

```
>> + } else {
```

```
>> + if (++sig->posix_timer_id < 0)
```

```
>> + sig->posix_timer_id = 0;
```

```
>> + if (sig->posix_timer_id == next_free_id)
```

```
>> + ret = -EAGAIN;
```

```
>> + }
```

```
>> + spin_unlock(&hash_lock);
```

```
>> + } while (ret == -ENOENT);
```

```
>> + return ret;
```

```
>> +}
```

```
>> +
```

>

> You probably need to add a rcu\_assign\_pointer() or smp\_wmb() before  
> the :

>

```
> new_timer->it_signal = current->signal;
```

>

> in the following block :

>

```
> spin_lock_irq(&current->sigband->siglock);
```

```
> new_timer->it_signal = current->signal;
```

```
> list_add(&new_timer->list, &current->signal->posix_timers);
```

```
> spin_unlock_irq(&current->sigband->siglock);
```

>

> Or else another thread can read outdated informations...

```
>  
>  
>     spin_lock_irq(&current->sighand->siglock);  
>     list_add(&new_timer->list, &current->signal->posix_timers);  
>     spin_unlock_irq(&current->sighand->siglock);  
> smp_wmb();  
>     new_timer->it_signal = current->signal;  
>  
>
```

Sorry, but I don't understand what races you are talking about.  
Could you specify, please?

--

Best regards,  
Stanislav Kinsbursky

---