

[patch-stable 3/3] pi-futex: Fix exit races and locking problems

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-06/msg03253.html>

- *From:* Thomas Gleixner <tglx@xxxxxxxxxxxxxx>
 - *Date:* Fri, 08 Jun 2007 10:29:30 -0000
-

From: Alexey Kuznetsov <kuznet@xxxxxxxxxxxxxx>

1. New entries can be added to `tsk->pi_state_list` after task completed `exit_pi_state_list()`. The result is memory leakage and deadlocks.

2. `handle_mm_fault()` is called under spinlock. The result is obvious.

3. results in self-inflicted deadlock inside `glibc`.

Sometimes `futex_lock_pi` returns `-ESRCH`, when it is not expected and `glibc` enters to `for(;;) sleep()` to simulate deadlock. This problem is quite obvious and I think the patch is right. Though it looks like each "if" in `futex_lock_pi()` got some stupid special case "else if". :-)

4. sometimes `futex_lock_pi()` returns `-EDEADLK`, when nobody has the lock. The reason is also obvious (see comment in the patch), but correct fix is far beyond my comprehension. I guess someone already saw this, the chunk:

```
if (rt_mutex_trylock(&q.pi_state->pi_mutex))
ret = 0;
```

is obviously from the same opera. But it does not work, because the `rtmutex` is really taken at this point: `wake_futex_pi()` of previous owner reassigned it to us. My fix works. But it looks very stupid. I would think about removal of shift of ownership in `wake_futex_pi()` and making all the work in context of process taking lock.

From: Thomas Gleixner <tglx@xxxxxxxxxxxxxx>

Fix 1) Avoid the tasklist lock variant of the exit race fix by adding an additional state transition to the exit code.

This fixes also the issue, when a task with recursive segfaults is not able to release the futexes.

Fix 2) Cleanup the `lookup_pi_state()` failure path and solve the `-ESRCH` problem finally.

[patch-stable 3/3] pi-futex: Fix exit races and locking problems

Fix 3) Solve the fixup_pi_state_owner() problem which needs to do the fixup in the lock protected section by using the in_atomic userspace access functions.

This removes also the ugly lock drop / unqueue inside of fixup_pi_state()

Fix 4) Fix a stale lock in the error path of futex_wake_pi()

Added some error checks for verification.

The -EDEADLK problem is solved by the rtmutex fixups.

Signed-off-by: Thomas Gleixner <tglx@xxxxxxxxxxxxxx>

Acked-by: Ingo Molnar <mingo@xxxxxxx>

include/linux/sched.h | 1
kernel/exit.c | 24 ++++++
kernel/futex.c | 191 +++-----
3 files changed, 151 insertions(