Subject: [PATCH -mm] task_struct: move ->fpu_counter and ->oomkilladj
Posted by Alexey Dobriyan on Thu, 27 Sep 2007 09:55:57 GMT
View Forum Message <> Reply to Message

There is nice 2 byte hole after struct task_struct::ioprio field
into which we can put two 1-byte fields: ->fpu_counter and ->oomkilladj.

[cc'ing Arjan just in case ->fpu_counter placement wasn't completely random :^)]

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>
---

 include/linux/sched.h | 20 ++++++++++----------
 1 file changed, 10 insertions(+), 10 deletions(-)

--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -960,6 +960,16 @@ struct task_struct {
 #endif

  unsigned short ioprio;
+ /*
+ * fpu_counter contains the number of consecutive context switches
+ * that the FPU is used. If this is over a threshold, the lazy fpu
+ * saving becomes unlazy to save the trap. This is an unsigned char
+ * so that after 256 times the counter wraps and the behavior turns
+ * lazy again; this to deal with bursty apps that only use FPU for
+ * a short time
+ */
+ unsigned char fpu_counter;
+ s8 oomkilladj; /* OOM kill score adjustment (bit shift). */
 #ifdef CONFIG_BLK_DEV_IO_TRACE
  unsigned int btrace_seq;
 #endif
@@ -1044,16 +1054,6 @@ struct task_struct {
  struct key *thread_keyring; /* keyring private to this thread */
  unsigned char jit_keyring; /* default keyring to attach requested keys to */
 #endif
- /*
- * fpu_counter contains the number of consecutive context switches
- * that the FPU is used. If this is over a threshold, the lazy fpu
- * saving becomes unlazy to save the trap. This is an unsigned char
- * so that after 256 times the counter wraps and the behavior turns
- * lazy again; this to deal with bursty apps that only use FPU for
- * a short time
- */
- unsigned char fpu_counter;
- s8 oomkilladj; /* OOM kill score adjustment (bit shift). */

```
char comm[TASK_COMM_LEN]; /* executable name excluding path
        - access with [gs]et_task_comm (which lock
          it with task_lock())
```