

[patch 1/2] hrtimer

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-03/msg08818.html>

- *From:* Thomas Gleixner <tglx@xxxxxxxxxxxxxx>
 - *Date:* Sat, 25 Mar 2006 12:46:01 -0000
-

The removal of the data field in the hrtimer structure enforces the embedding of the timer into another data structure. nanosleep now uses a private implementation of the most common used timer callback function (simple task wakeup). In order to avoid the reimplement of such functionality all over the place a generic hrtimer_sleeper functionality is created.

Signed-off-by: Thomas Gleixner <tglx@xxxxxxxxxxxxxx>
Signed-off-by: Ingo Molnar <mingo@xxxxxxx>

include/linux/hrtimer.h | 16 +++++
kernel/hrtimer.c | 19 +++++
2 files changed, 35 insertions(+)

Index: linux-2.6.16/include/linux/hrtimer.h

```
=====
--- linux-2.6.16.orig/include/linux/hrtimer.h
+++ linux-2.6.16/include/linux/hrtimer.h
@@ -58,6 +58,19 @@ struct hrtimer {
};

/**
+ * struct hrtimer_sleeper - simple sleeper structure
+ *
+ * @timer: embedded timer structure
+ * @task: task to wake up
+ *
+ * task is set to NULL, when the timer expires.
+ */
+struct hrtimer_sleeper {
+ struct hrtimer timer;
+ struct task_struct *task;
+};
+
+/**
* struct hrtimer_base - the timer base for a specific clock
*
* @index: clock type index for per_cpu support when moving a timer
@@ -127,6 +140,9 @@ extern long hrtimer_nanosleep(struct tim
```

[patch 1/2] hrtimer

```
const enum hrtimer_mode mode,
const clockid_t clockid);

+extern void hrtimer_init_sleeper(struct hrtimer_sleeper *sl,
+ struct task_struct *tsk);
+
+/* Soft interrupt function to run the hrtimer queues: */
extern void hrtimer_run_queues(void);
```

Index: linux-2.6.16/kernel/hrtimer.c

```
----- linux-2.6.16.orig/kernel/hrtimer.c
+++ linux-2.6.16/kernel/hrtimer.c
@@ -656,6 +656,25 @@ void hrtimer_run_queues(void)
 * Sleep related functions:
 */

+static int hrtimer_wakeup(struct hrtimer *timer)
+{
+ struct hrtimer_sleeper *t =
+ container_of(timer, struct hrtimer_sleeper, timer);
+ struct task_struct *task = t->task;
+
+ t->task = NULL;
+ if (task)
+ wake_up_process(task);
+
+ return HRTIMER_NORESTART;
+}
+
+void hrtimer_init_sleeper(struct hrtimer_sleeper *sl, task_t *task)
+{
+ sl->timer.function = hrtimer_wakeup;
+ sl->task = task;
+}
+
+struct sleep_hrtimer {
+ struct hrtimer timer;
+ struct task_struct *task;
+
+-----
+
+-----
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the