
Subject: [PATCH, v6 2/3] Implement timer slack notifier chain
Posted by [Kirill A. Shutemov](#) on Mon, 14 Feb 2011 13:06:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Kirill A. Shutemov <kirill@shutemov.name>

Process can change its timer slack using prctl(). Timer slack notifier call chain allows to react on such change or forbid it.

Signed-off-by: Kirill A. Shutemov <kirill@shutemov.name>

```
include/linux/sched.h | 5 ++++
kernel/Makefile       | 2 +-
kernel/sys.c          | 9 +-----
kernel/timer_slack.c  | 57 ++++++
4 files changed, 65 insertions(+), 8 deletions(-)
create mode 100644 kernel/timer_slack.c
```

```
diff --git a/include/linux/sched.h b/include/linux/sched.h
index d747f94..f1a4ec2 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -2620,6 +2620,11 @@ static inline unsigned long rlimit_max(unsigned int limit)
    return task_rlimit_max(current, limit);
}
```

```
+extern int register_timer_slack_notifier(struct notifier_block *nb);
+extern void unregister_timer_slack_notifier(struct notifier_block *nb);
+long prctl_get_timer_slack(void);
+extern long prctl_set_timer_slack(long timer_slack_ns);
+
#endif /* __KERNEL__ */
```

```
#endif
diff --git a/kernel/Makefile b/kernel/Makefile
index 353d3fe..aa43e63 100644
--- a/kernel/Makefile
+++ b/kernel/Makefile
@@ -10,7 +10,7 @@ obj-y = sched.o fork.o exec_domain.o panic.o printk.o \
    kthread.o wait.o kfifo.o sys_ni.o posix-cpu-timers.o mutex.o \
    hrtimer.o rwsem.o nsproxy.o srcu.o semaphore.o \
    notifier.o ksysfs.o pm_qos_params.o sched_clock.o cred.o \
-   async.o range.o jump_label.o
+   async.o range.o jump_label.o timer_slack.o
obj-y += groups.o
```

```
ifdef CONFIG_FUNCTION_TRACER
diff --git a/kernel/sys.c b/kernel/sys.c
```

```

index 18da702..a33e4b8 100644
--- a/kernel/sys.c
+++ b/kernel/sys.c
@@ -1691,15 +1691,10 @@ SYSCALL_DEFINE5(prctl, int, option, unsigned long, arg2,
unsigned long, arg3,
    error = perf_event_task_enable();
    break;
    case PR_GET_TIMERSLACK:
-   error = current->timer_slack_ns;
+   error = prctl_get_timer_slack();
    break;
    case PR_SET_TIMERSLACK:
-   if (arg2 <= 0)
-       current->timer_slack_ns =
-       current->default_timer_slack_ns;
-   else
-       current->timer_slack_ns = arg2;
-   error = 0;
+   error = prctl_set_timer_slack(arg2);
    break;
    case PR_MCE_KILL:
        if (arg4 | arg5)
diff --git a/kernel/timer_slack.c b/kernel/timer_slack.c
new file mode 100644
index 0000000..546ed78
--- /dev/null
+++ b/kernel/timer_slack.c
@@ -0,0 +1,57 @@
+/*
+ * kernel/timer_slack.c
+ *
+ * Copyright Nokia Corporation, 2011
+ * Author: Kirill A. Shutemov
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ */
+
+#include <linux/notifier.h>
+#include <linux/module.h>
+#include <linux/sched.h>

```

```

+
+static BLOCKING_NOTIFIER_HEAD(timer_slack_notify_list);
+
+int register_timer_slack_notifier(struct notifier_block *nb)
+{
+ return blocking_notifier_chain_register(&timer_slack_notify_list, nb);
+}
+EXPORT_SYMBOL_GPL(register_timer_slack_notifier);
+
+void unregister_timer_slack_notifier(struct notifier_block *nb)
+{
+ blocking_notifier_chain_unregister(&timer_slack_notify_list, nb);
+}
+EXPORT_SYMBOL_GPL(unregister_timer_slack_notifier);
+
+long prctl_get_timer_slack(void)
+{
+ return current->timer_slack_ns;
+}
+
+long prctl_set_timer_slack(long timer_slack_ns)
+{
+ int err;
+
+ /* Reset timer slack to default value */
+ if (timer_slack_ns <= 0) {
+ current->timer_slack_ns = current->default_timer_slack_ns;
+ return 0;
+ }
+
+ err = blocking_notifier_call_chain(&timer_slack_notify_list,
+ timer_slack_ns, NULL);
+ if (err == NOTIFY_DONE)
+ current->timer_slack_ns = timer_slack_ns;
+
+ return notifier_to_errno(err);
+}
--

```

1.7.4

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
