
Subject: Re: [PATCH, v6 2/3] Implement timer slack notifier chain
Posted by [Thomas Gleixner](#) on Mon, 14 Feb 2011 13:32:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 14 Feb 2011, Kirill A. Shutsemov wrote:

> 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.

So we add a notifier call chain and more exports to allow what ?

```
> --- 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();
```

What's the point of replacing current->timer_slack_ns with a function which does exactly the same ?

```
> +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;
```

That does not make any sense at all. Why is setting default_timer_slack_ns not subject to validation ? Why is it treaded seperately ?

```
> + }
> +
> + 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);
```

Thanks,

tglx

Containers mailing list

Containers@lists.linux-foundation.org

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