
Subject: Re: [RFC PATCH 3/6] IPC/sem: start/stop operations for /proc/pid/semundo

Posted by [serue](#) on Wed, 25 Jun 2008 20:39:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quoting Nadia.Derbey@bull.net (Nadia.Derbey@bull.net):

> PATCH [03/06]

>

> This patch introduces the .start and .stop seq operations for

> /proc/pid/semundo.

>

> Signed-off-by: Pierre Peiffer <pierre.peiffer@bull.net>

> Signed-off-by: Nadia Derby <Nadia.Derbey@bull.net>

Acked-by: Serge Hallyn <serue@us.ibm.com>

> ---

> ipc/sem.c | 43 ++++++

> 1 file changed, 41 insertions(+), 2 deletions(-)

>

> Index: linux-2.6.26-rc5-mm3/ipc/sem.c

> =====

> --- linux-2.6.26-rc5-mm3.orig/ipc/sem.c 2008-06-24 10:59:46.000000000 +0200

> +++ linux-2.6.26-rc5-mm3/ipc/sem.c 2008-06-24 12:32:36.000000000 +0200

> @@ -1400,7 +1400,42 @@ struct undo_list_data {

> /* iterator */

> static void *semundo_start(struct seq_file *m, loff_t *ppos)

> {

> - return NULL;

> + struct undo_list_data *data = m->private;

> + struct sem_undo_list *ulp = data->undo_list;

> + struct sem_undo *undo;

> + loff_t pos = *ppos;

> +

> + if (!ulp)

> + return NULL;

> +

> + if (pos < 0)

> + return NULL;

> +

> + /* If ulp is not NULL, it means that we've successfully grabbed

> + * a refcnt in semundo_open. That prevents the undo_list from being

> + * freed.

> + *

> + * Note:

> + * 1) the sem_undo structure is RCU-protected. So take the rcu read

> + * lock and only release it during the .stop operation.

> + * 2) we accept to release the undo_list lock (i.e. we allow the list

```

> + *   to change) between each iteration, instead of taking that lock
> + *   during the .start and releasing it during the .stop operation.
> + *   This is to reduce the performance impact on the access to the
> + *   undo_list.
> + */
> + rcu_read_lock();
> + spin_lock(&ulp->lock);
> + list_for_each_entry_rcu(undo, &ulp->list_proc, list_proc) {
> +   if ((undo->semid != -1) && !(pos--))
> +     break;
> + }
> + spin_unlock(&ulp->lock);
> +
> + if (&undo->list_proc == &ulp->list_proc)
> +   return NULL;
> +
> + return undo;
> }
>
> static void *semundo_next(struct seq_file *m, void *v, loff_t *ppos)
> @@ -1410,7 +1445,11 @@ static void *semundo_next(struct seq_fil
>
> static void semundo_stop(struct seq_file *m, void *v)
> {
> - return;
> + struct undo_list_data *data = m->private;
> + struct sem_undo_list *ulp = data->undo_list;
> +
> + if (ulp)
> +   rcu_read_unlock();
> }
>
> static int semundo_show(struct seq_file *m, void *v)
>
> --

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
