Subject: Re: [PATCH 8/8] Re-enable msgmni automatic recomputing msgmni if set to negative
Posted by Nadia Derbey on Tue, 12 Feb 2008 11:38:38 GMT
View Forum Message <> Reply to Message

Andrew Morton wrote:
> On Mon, 11 Feb 2008 15:16:54 +0100
> Nadia.Derbey@bull.net wrote:
>
>
>>[PATCH 08/08]
>>
>>This patch is the enhancement as asked for by Yasunori: if msgmni is set to
>>a negative value, register it back into the ipcns notifier chain.
>>
>>A new interface has been added to the notification mechanism:
>>notifier_chain_cond_register() registers a notifier block only if not already
>>registered. With that new interface we avoid taking care of the states changes
>>in procfs.
>>
>>...
>>
>> static int proc_ipc_callback_dointvec(ctl_table *table, int write,
>> struct file *filp, void __user *buffer, size_t *lenp, loff_t *ppos)
>> {
>>+ struct ctl_table ipc_table;
>> size_t lenp_bef = *lenp;
>> int rc;
>>
>>- rc = proc_ipc_dointvec(table, write, filp, buffer, lenp, ppos);
>>+ memcpy(&ipc_table, table, sizeof(ipc_table));
>>+ ipc_table.data = get_ipc(table);
>>+
>>+ rc = proc_dointvec(&ipc_table, write, filp, buffer, lenp, ppos);
>>
>> if (write && !rc && lenp_bef == *lenp)
>>- /*
>>-   * Tunable has successfully been changed from userland:
>>-   * disable its automatic recomputing.
>>-   */
>>- unregister_ipcns_notifier(current->nsproxy->ipc_ns);
>>+ tunable_set_callback(*((int *)(ipc_table.data)));
>>
>> return rc;
>> }
>>@@ -119,12 +142,14 @@ static int sysctl_ipc_registered_data(ct
>> rc = sysctl_ipc_data(table, name, nlen, oldval, oldlenp, newval,
>>   newlen);

>>
>>- if (newval && newlen && rc > 0)
>>+ if (newval && newlen && rc > 0) {
>> /*
>>- * Tunable has successfully been changed from userland:
>>- * disable its automatic recomputing.
>>+ * Tunable has successfully been changed from userland
>>   */
>>- unregister_ipcns_notifier(current->nsproxy->ipc_ns);
>>+ int *data = get_ipc(table);
>>+
>>+ tunable_set_callback(*data);
>>+ }
>>
>> return rc;
>> }
>
>
> hm, what's happening here? We take a local copy of the caller's ctl_table
> and then pass that into proc_dointvec(). Is that as hacky as it seems??
>
>

Well, the caller's ctl_table contains the tunables addresses for the
init namspeace in its .data fields. While what needs to be passed in to
proc_dointvec() is the tunable address in the caller's namespace.
Since all the fields in ipc_kern_table[] are ok but the .data one, imho
it's correct to store it in a local copy and change the data field to
the appropirate one, before passing it to proc_dointvec().

Regards,
Nadia

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