Subject: Re: [PATCH 01/11] SYSCTL: export root and set handling routines
Posted by Stanislav Kinsbursky on Mon, 19 Dec 2011 08:56:52 GMT
View Forum Message <> Reply to Message

> Stanislav Kinsbursky<skinsbursky@parallels.com>  writes:
>
>> These routines are required for making SUNRPC sysctl's per network namespace
>> context.
>
> Why does sunrpc require it's own sysctl root?  You should be able to use
> the generic per network namespace root and call it good.
>
> What makes register_net_sysctl_table and register_net_sysctl_ro_table
> unsuitable for sunrpc.  I skimmed through your patches and I haven't
> seen anything obvious.
>
> Eric
>

Hello, Eric. Sorry for the lack of information.
I was considering two ways how to make these sysctl per net ns:

1) Use register_net_sysctl_table and register_net_sysctl_ro_table as you mentioned. This was easy and cheap, but also means, than all user-space programs, tuning SUNRPC will be broken (since all sysctl currently located in"/proc/sys/sunprc/").

2) Export sysctl root creation routines and make per-net SUNRPC sysctl root. This approach allows to make any part of sysctl tree per namespace context and thus leave user-space stuff unchanged.

BTW, NFS and LockD also have it's sysctls ("/proc/sys/fs/nfs/").
And also because of them I've decided, that it would be better to export SYSCTL root creation routines instead of breaking compatibility for all NFS layers by moving all sysctl under /proc/sys/net/ directory.

Do you feel that it was a bad decision?

>
>> Signed-off-by: Stanislav Kinsbursky<skinsbursky@parallels.com>
>>
>> ---
>>  include/linux/sysctl.h |    1 +
>>  kernel/sysctl.c        |   11 ++++++++++
>>  2 files changed, 12 insertions(+), 0 deletions(-)
>>
>> diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h

```
>> index 703cfa3..be586a9 100644
>> --- a/include/linux/sysctl.h
>> +++ b/include/linux/sysctl.h
>> @@ -1084,6 +1084,7 @@ struct ctl_path {
>>   };
>>
>>   void register_sysctl_root(struct ctl_table_root *root);
>> +void unregister_sysctl_root(struct ctl_table_root *root);
>>   struct ctl_table_header *__register_sysctl_paths(
>>    struct ctl_table_root *root, struct nsproxy *namespaces,
>>    const struct ctl_path *path, struct ctl_table *table);
>> diff --git a/kernel/sysctl.c b/kernel/sysctl.c
>> index ae27196..fb016a9 100644
>> --- a/kernel/sysctl.c
>> +++ b/kernel/sysctl.c
>> @@ -1700,6 +1700,15 @@ void register_sysctl_root(struct ctl_table_root *root)
>>    list_add_tail(&root->root_list,&sysctl_table_root.root_list);
>>    spin_unlock(&sysctl_lock);
>>   }
>> +EXPORT_SYMBOL_GPL(register_sysctl_root);
>> +
>> +void unregister_sysctl_root(struct ctl_table_root *root)
>> +{
>> + spin_lock(&sysctl_lock);
>> + list_del(&root->root_list);
>> + spin_unlock(&sysctl_lock);
>> +}
>> +EXPORT_SYMBOL_GPL(unregister_sysctl_root);
>>
>>   /*
>>    * sysctl_perm does NOT grant the superuser all rights automatically, because
>> @@ -1925,6 +1934,7 @@ struct ctl_table_header *__register_sysctl_paths(
>>
>>    return header;
>>   }
>> +EXPORT_SYMBOL_GPL(__register_sysctl_paths);
>>
>>   /**
>>    * register_sysctl_table_path - register a sysctl table hierarchy
>> @@ -2007,6 +2017,7 @@ void setup_sysctl_set(struct ctl_table_set *p,
>>    p->parent = parent ? parent :&sysctl_table_root.default_set;
>>    p->is_seen = is_seen;
>>   }
>> +EXPORT_SYMBOL_GPL(setup_sysctl_set);
>>
>>   #else /* !CONFIG_SYSCTL */
>>   struct ctl_table_header *register_sysctl_table(struct ctl_table * table)
>>
```

>> --
>> To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
>> the body of a message to majordomo@vger.kernel.org
>> More majordomo info at  http://vger.kernel.org/majordomo-info.html
>> Please read the FAQ at  http://www.tux.org/lkml/
> --
> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at  http://vger.kernel.org/majordomo-info.html

--
Best regards,
Stanislav Kinsbursky

---