Subject: namespaces compatibility list

Posted by Pavel Emelianov on Tue, 06 Nov 2007 10:51:21 GMT

View Forum Message <> Reply to Message

Hi guys!

As you might have seen, recently there was some spontaneous
discussion about the namespaces-working-together problems.

Ted T'so proposed to create some document that describes what
problems user may have when he/she creates some new namespace,
but keeps others shared. I like this idea, so here's the draft
with the problems I currently have in mind and can describe
somewhat audibly - the "namespaces compatibility list".

The Documentation/namespaces/ dir is about to contain more
docs about the namespaces stuff (e.g. I'm going to prepare
a doc about the pid namespaces, maybe Serge will want to
write something about the user namespaces development, Eric
may want to put some notes about the netns API and so on),
but currently there will be only one file.

What would you say about it?

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

diff --git a/Documentation/namespaces/compatibility-list.txt
b/Documentation/namespaces/compatibility-list.txt
new file mode 100644
index 0000000..4be4a3c
--- /dev/null
+++ b/Documentation/namespaces/compatibility-list.txt
@@ -0,0 +1,32 @@
+ Namespaces compatibility list
+
+This document contains the information about the problems user
+may have when creating tasks living in different namespaces.
+
+Here's the summary. This matrix shows the known problems, that
+occur when tasks share some namespace (the columns) while living
+in different other namespaces (the raws):
+
+ UTS IPC VFS PID User Net
+UTS
+IPC   1
+VFS

```
+PID   1 1
+User    2
+Net
+
+1. Both the IPC and the PID namespaces provide IDs to address
+   object inside the kernel. E.g. semaphore with ipcid or
+   process group with pid.
+
+   In both cases, tasks shouldn't try telling this id to some
+   other task living in different namespace vid shared filesystem
+   or IPC shmem/message. The fact is that this ID is only valid
+   within the namespace it was obtained in and may refer to some
+   other object in another namespace.
+
+2. Intentionnaly, two equal user ids in different user namespaces
+   should not be equal from the VFS point of view. In other
+   words, user 10 in one user namespace shouldn't have the same
+   access permissions to files, beloging to user 10 in another
+   namespace. But currently this is not so.
```

_____