
Subject: Re: - merge-sys_clone-sys_unshare-nsproxy-and-namespace.patch
removed from -mm tree

Posted by [Herbert Poetzl](#) on Mon, 18 Jun 2007 15:41:08 GMT

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

On Mon, Jun 18, 2007 at 02:02:19PM +0200, Cedric Le Goater wrote:

>
> > on Linux-VServer, we have accounting for those
> > proxies (and several other namespace related stuff)
> > because we already suspected leakage and reference
> > bugs in this area some time ago ... btw, I also
> > suggested to put a similar functionality in mainline
> > for the time being, but it was ignored, as usual ...
>
> something like a kmem_cache ?

maybe, but even simplest accounting of the
different 'objects' would be more than enough
for the test phase (or maybe as statistics :)

> and we are not ignoring vservers ! :)

good for them, our project is called Linux-VServer :)

best,
Herbert

> Cheers,
>
> C.
>
> Add a kmem_cache to manage nsproxy objects.
>
> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
> ---

> kernel/nsproxy.c | 21 ++++++++-----
> 1 file changed, 17 insertions(+), 4 deletions(-)
>

> Index: 2.6.22-rc4-mm2/kernel/nsproxy.c

> =====

> --- 2.6.22-rc4-mm2.orig/kernel/nsproxy.c

> +++ 2.6.22-rc4-mm2/kernel/nsproxy.c

> @@ -21,6 +21,8 @@

> #include <linux/utsname.h>

> #include <linux/pid_namespace.h>

>

> +static struct kmem_cache *nsproxy_cachep;

> +

```

> struct nsproxy init_nsproxy = INIT_NS_PROXY(init_nsproxy);
>
> static inline void get_nsproxy(struct nsproxy *ns)
@@ -43,9 +45,11 @@ static inline struct nsproxy *clone_nspr
> {
>     struct nsproxy *ns;
>
>     ns = kmemdup(orig, sizeof(struct nsproxy), GFP_KERNEL);
>     if (ns)
>         ns = kmem_cache_alloc(nsproxy_cache, GFP_KERNEL);
>     if (ns) {
>         memcpy(ns, orig, sizeof(struct nsproxy));
>         atomic_set(&ns->count, 1);
>     }
>     return ns;
> }
>
@@ -109,7 +113,7 @@ out_uts:
>     if (new_nsp->mnt_ns)
>         put_mnt_ns(new_nsp->mnt_ns);
>     out_ns:
>     kfree(new_nsp);
>     kmem_cache_free(nsproxy_cache, new_nsp);
>     return ERR_PTR(err);
> }
>
@@ -160,7 +164,7 @@ void free_nsproxy(struct nsproxy *ns)
>     put_pid_ns(ns->pid_ns);
>     if (ns->user_ns)
>         put_user_ns(ns->user_ns);
>     kfree(ns);
>     kmem_cache_free(nsproxy_cache, ns);
> }
>
> /*
@@ -191,3 +195,12 @@ int unshare_nsproxy_namespaces(unsigned
> }
>     return err;
> }
>
> +static int __init nsproxy_cache_init(void)
> +{
>     nsproxy_cache = kmem_cache_create("nsproxy", sizeof(struct nsproxy),
>     0, SLAB_PANIC, NULL, NULL);
>     return 0;
> }
>
> +module_init(nsproxy_cache_init);

```

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