Subject: Re: [PATCH 5/5] Move alloc_pid call to copy_process
Posted by Oleg Nesterov on Sun, 15 Jul 2007 12:36:00 GMT
View Forum Message <> Reply to Message

Sukadev Bhattiprolu wrote:
>
> --- lx26-22-rc6-mm1.orig/kernel/pid.c 2007-07-13 18:23:55.000000000 -0700
> +++ lx26-22-rc6-mm1/kernel/pid.c 2007-07-13 18:23:55.000000000 -0700
> @@ -206,6 +206,10 @@ fastcall void free_pid(struct pid *pid)
> /* We can be called with write_lock_irq(&tasklist_lock) held */
> unsigned long flags;
>
> + /* check this here to keep copy_process() cleaner */
> + if (unlikely(pid == &init_struct_pid))
> + return;
> +
> spin_lock_irqsave(&pidmap_lock, flags);
> hlist_del_rcu(&pid->pid_chain);
> spin_unlock_irqrestore(&pidmap_lock, flags);
> @@ -214,13 +218,17 @@ fastcall void free_pid(struct pid *pid)
> call_rcu(&pid->rcu, delayed_put_pid);
> }
>
> -struct pid *alloc_pid(void)
> +struct pid *alloc_pid(enum copy_process_type copy_src)
> {
> struct pid *pid;
> enum pid_type type;
> int nr = -1;
> struct pid_namespace *ns;
>
> + /* check this here to keep copy_process() cleaner */
> + if (unlikely(copy_src == COPY_IDLE_PROCESS))
> + return &init_struct_pid;
> +
> ns = task_active_pid_ns(current);
> pid = kmem_cache_alloc(ns->pid_cachep, GFP_KERNEL);
> if (!pid)

Ugh. I am sorry! but imho this is so ugly :(

Could you please give more details why we need this change? Even if we really need this, can't we do these checks in copy_process() ? free_pid(&init_struct_pid) can only happen when fork fails, this is a slow path, and copy_process() knows copy_src.

Oleg.

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