
Subject: Re: [PATCH 8/15] Helpers to find the task by its numerical ids
Posted by [Dave Hansen](#) on Thu, 26 Jul 2007 19:05:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 2007-07-26 at 18:51 +0400, Pavel Emelyanov wrote:

```
> +extern struct task_struct *find_task_by_pid_type_ns(int type, int pid,  
> + struct pid_namespace *ns);  
> +  
> +#define find_task_by_pid_ns(nr, ns) \  
> + find_task_by_pid_type_ns(PIDTYPE_PID, nr, ns)  
> +#define find_task_by_pid_type(type, nr) \  
> + find_task_by_pid_type_ns(type, nr, &init_pid_ns)  
> +#define find_task_by_pid(nr) \  
> + find_task_by_pid_type(PIDTYPE_PID, nr)  
> +  
> extern void __set_special_pids(pid_t session, pid_t pgrp);
```

Do these `_have_` to be macros?

```
> /* per-UID process charging. */  
> diff -upr linux-2.6.23-rc1-mm1.orig/kernel/pid.c linux-2.6.23-rc1-mm1-7/kernel/pid.c  
> --- linux-2.6.23-rc1-mm1.orig/kernel/pid.c 2007-07-26 16:34:45.000000000 +0400  
> +++ linux-2.6.23-rc1-mm1-7/kernel/pid.c 2007-07-26 16:36:37.000000000 +0400  
> @@ -204,19 +221,20 @@ static void delayed_put_pid(struct rcu_h  
> goto out;  
> }  
>   
> -struct pid *fastcall find_pid(int nr)  
> +struct pid *fastcall find_pid_ns(int nr, struct pid_namespace *ns)  
> {  
> struct hlist_node *elem;  
> - struct pid *pid;  
> + struct upid *pnr;  
> +  
> + hlist_for_each_entry_rcu(pnr, elem,  
> + &pid_hash[pid_hashfn(nr, ns)], pid_chain)  
> + if (pnr->nr == nr && pnr->ns == ns)  
> + return container_of(pnr, struct pid,  
> + numbers[ns->level]);
```

Do we do this loop anywhere else? Should we have a `for_each_pid()` that makes this a little less messy?

```
> - hlist_for_each_entry_rcu(pid, elem,  
> - &pid_hash[pid_hashfn(nr)], pid_chain) {  
> - if (pid->nr == nr)  
> - return pid;  
> - }
```

```

> return NULL;
> }
> -EXPORT_SYMBOL_GPL(find_pid);
> +EXPORT_SYMBOL_GPL(find_pid_ns);
>
> /*
>  * attach_pid() must be called with the tasklist_lock write-held.
>  @@ -318,12 +355,13 @@ struct task_struct *fastcall pid_task(s
>  /*
>  * Must be called under rcu_read_lock() or with tasklist_lock read-held.
>  */
> -struct task_struct *find_task_by_pid_type(int type, int nr)
> +struct task_struct *find_task_by_pid_type_ns(int type, int nr,
> + struct pid_namespace *ns)
> {
> - return pid_task(find_pid(nr), type);
> + return pid_task(find_pid_ns(nr, ns), type);
> }
>
> -EXPORT_SYMBOL(find_task_by_pid_type);
> +EXPORT_SYMBOL(find_task_by_pid_type_ns);
>
> struct pid *get_task_pid(struct task_struct *task, enum pid_type type)
> {
> @@ -342,7 +426,7 @@ struct pid *find_get_pid(pid_t nr)
>  struct pid *pid;
>
>  rcu_read_lock();
> - pid = get_pid(find_pid(nr));
> + pid = get_pid(find_vpid(nr));
>  rcu_read_unlock();

```

OK, I think this is really confusing. If find_get_pid() finds vpids, should we not call it find_get_vpid()?

-- Dave
