
Subject: Re: [RFC][PATCH 4/4] PID: use the target ID specified in procs

Posted by [Pavel Emelianov](#) on Tue, 11 Mar 2008 12:04:39 GMT

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

Nadia.Derbey@bull.net wrote:

```
> @@ -122,14 +122,26 @@ static void free_pidmap(struct upid *upi
>   atomic_inc(&map->nr_free);
> }
>
> -static int alloc_pidmap(struct pid_namespace *pid_ns)
> +static int alloc_pidmap(struct pid_namespace *pid_ns, struct pid_list *pid_l,
> +   int level)
> {
>   int i, offset, max_scan, pid, last = pid_ns->last_pid;
>   struct pidmap *map;
>
>   - pid = last + 1;
>   - if (pid >= pid_max)
>   -   pid = RESERVED_PIDS;
>   + if (!pid_l) {
>   +   pid = last + 1;
>   +   if (pid >= pid_max)
>   +     pid = RESERVED_PIDS;
>   + } else {
>   +   /*
>   +    * There's a target pid, so use it instead
>   +    */
>   +   BUG_ON(level < 0);
>   +   pid = PID_AT(pid_l, level);
>   +   if (pid >= pid_max)
>   +     return -EINVAL;
>   + }
>   +
>   offset = pid & BITS_PER_PAGE_MASK;
>   map = &pid_ns->pidmap[pid/BITS_PER_PAGE];
>   max_scan = (pid_max + BITS_PER_PAGE - 1)/BITS_PER_PAGE - !offset;
> @@ -153,9 +165,16 @@ static int alloc_pidmap(struct pid_names
>   do {
>     if (!test_and_set_bit(offset, map->page)) {
>       atomic_dec(&map->nr_free);
>       -   pid_ns->last_pid = pid;
>       +   if (!pid_l)
>       +     pid_ns->last_pid = pid;
>       +   else
>       +     pid_ns->last_pid = max(last,
>       +       pid);
>       return pid;
>     }
>   }
```

```

> + if (pid_l)
> + /* Target pid is already in use */
> + return -EBUSY;
> offset = find_next_offset(map, offset);
> pid = mk_pid(pid_ns, map, offset);
> /*
> @@ -179,7 +198,7 @@ static int alloc_pidmap(struct pid_names
> }
> pid = mk_pid(pid_ns, map, offset);
> }
> - return -1;
> + return -ENOMEM;
> }
>
> int next_pidmap(struct pid_namespace *pid_ns, int last)

```

As far as this particular piece of code is concerned this all can be shrunk down to

```

static int set_vpidmap(struct pid_namespace *ns, int pid)
{
    int offset;
    pidmap_t *map;

    offset = pid & BITS_PER_PAGE_MASK;
    map = ns->pidmap + vpid / BITS_PER_PAGE;

    if (unlikely(alloc_pidmap_page(map)))
        return -ENOMEM;

    if (test_and_set_bit(offset, map->page))
        return -EEXIST;

    atomic_dec(&map->nr_free);
    return pid;
}

```

where the `alloc_pidmap_page` is a consolidated part of code from `alloc_pidmap`.

And I'm scared of what the `alloc_pid` is going to become.

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