

---

Subject: [PATCH] Use task\_pid\_nr() instead of pid\_nr(task\_pid())  
Posted by [Pavel Emelianov](#) on Wed, 03 Oct 2007 14:26:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

There are two places that do so - the cgroups subsystem  
and the autofs code.

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

---

```
diff --git a/fs/autofs/root.c b/fs/autofs/root.c
index 592f640..5efff3c 100644
--- a/fs/autofs/root.c
+++ b/fs/autofs/root.c
@@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup

    oz_mode = autofs_oz_mode(sbi);
    DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
-   "oz_mode = %d\n", pid_nr(task_pid(current)),
+   "oz_mode = %d\n", task_pid_nr(current),
    task_pgrp_nr(current), sbi->catatonic,
    oz_mode));
```

```
diff --git a/kernel/cgroup.c b/kernel/cgroup.c
index 1e8aa53..8720881 100644
--- a/kernel/cgroup.c
+++ b/kernel/cgroup.c
@@ -1762,7 +1762,7 @@ static int pid_array_load(pid_t *pidarra
    while ((tsk = cgroup_iter_next(cont, &it)) {
        if (unlikely(n == npids))
            break;
-   pidarray[n++] = pid_nr(task_pid(tsk));
+   pidarray[n++] = task_pid_nr(tsk);
    }
    cgroup_iter_end(cont, &it);
    return n;
```

---

Subject: Re: [PATCH] Use task\_pid\_nr() instead of pid\_nr(task\_pid())  
Posted by [akpm](#) on Fri, 05 Oct 2007 23:15:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, 03 Oct 2007 18:26:06 +0400  
Pavel Emelyanov <xemul@openvz.org> wrote:

> There are two places that do so - the cgroups subsystem  
> and the autofs code.

```

>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> ---
>
> diff --git a/fs/autofs/root.c b/fs/autofs/root.c
> index 592f640..5efff3c 100644
> --- a/fs/autofs/root.c
> +++ b/fs/autofs/root.c
> @@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup
>
>  oz_mode = autofs_oz_mode(sbi);
>  DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
> -   "oz_mode = %d\n", pid_nr(task_pid(current)),
> +   "oz_mode = %d\n", task_pid_nr(current),
>    task_pgrp_nr(current), sbi->catatonic,
>    oz_mode));
>
> diff --git a/kernel/cgroup.c b/kernel/cgroup.c
> index 1e8aa53..8720881 100644
> --- a/kernel/cgroup.c
> +++ b/kernel/cgroup.c
> @@ -1762,7 +1762,7 @@ static int pid_array_load(pid_t *pidarra
>  while ((tsk = cgroup_iter_next(cont, &it)) {
>    if (unlikely(n == npids))
>      break;
> -   pidarray[n++] = pid_nr(task_pid(tsk));
> +   pidarray[n++] = task_pid_nr(tsk);
>  }
>  cgroup_iter_end(cont, &it);
>  return n;

```

And more will appear over time unless we remove pid\_nr() altogether.

Can we do that?

---

Subject: Re: [PATCH] Use task\_pid\_nr() instead of pid\_nr(task\_pid())

Posted by [Pavel Emelianov](#) on Mon, 08 Oct 2007 08:57:54 GMT

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

---

Andrew Morton wrote:

> On Wed, 03 Oct 2007 18:26:06 +0400

> Pavel Emelyanov <xemul@openvz.org> wrote:

>

>> There are two places that do so - the cgroups subsystem

>> and the autofs code.

>>

```

>> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>>
>> ---
>>
>> diff --git a/fs/autofs/root.c b/fs/autofs/root.c
>> index 592f640..5efff3c 100644
>> --- a/fs/autofs/root.c
>> +++ b/fs/autofs/root.c
>> @@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup
>>
>>  oz_mode = autofs_oz_mode(sbi);
>>  DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
>> -   "oz_mode = %d\n", pid_nr(task_pid(current)),
>> +   "oz_mode = %d\n", task_pid_nr(current),
>>    task_pgrp_nr(current), sbi->catatonic,
>>    oz_mode));
>>
>> diff --git a/kernel/cgroup.c b/kernel/cgroup.c
>> index 1e8aa53..8720881 100644
>> --- a/kernel/cgroup.c
>> +++ b/kernel/cgroup.c
>> @@ -1762,7 +1762,7 @@ static int pid_array_load(pid_t *pidarra
>>  while ((tsk = cgroup_iter_next(cont, &it))) {
>>    if (unlikely(n == npids))
>>      break;
>> -   pidarray[n++] = pid_nr(task_pid(tsk));
>> +   pidarray[n++] = task_pid_nr(tsk);
>>  }
>>  cgroup_iter_end(cont, &it);
>>  return n;
>
> And more will appear over time unless we remove pid_nr() altogether.
>
> Can we do that?

```

Hardly.

There are places that store the struct pid and try to obtain its number later. E.g. tiocpggrp() wants to tell user the pgrp nr of the tty's active group; fgetown() tells user the number of file's owner pid or (!) pgrp; file locks need to store the pid and report it later (this is not implemented yet, I'm still working on it) and so on and so forth.

Thanks,  
Pavel