
Subject: Re: [PATCH 03/10] Containers(V10): Add tasks file interface
Posted by [Cedric Le Goater](#) on Thu, 07 Jun 2007 14:00:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Paul !

menage@google.com wrote:

> This patch adds the per-directory "tasks" file for containerfs mounts;
> this allows the user to determine which tasks are members of a
> container by reading a container's "tasks", and to move a task into a
> container by writing its pid to its "tasks".

here's a small fix for 2.6.22-rc4-mm2.

C.

when there's no tasks in a container, opening

<containerfs>/<container>/tasks

spits the following warning because we are trying to
kmalloc(0).

```
WARNING: at mm/slab.c:777 __find_general_cachep()
[<c0102b01>] show_trace_log_lvl+0x1a/0x2f
[<c0103627>] show_trace+0x12/0x14
[<c010363e>] dump_stack+0x15/0x17
[<c0148040>] __kmalloc+0x56/0xf3
[<c012d030>] container_tasks_open+0x56/0x131
[<c012cdf8>] container_file_open+0x32/0x3a
[<c014964d>] __dentry_open+0x99/0x13c
[<c0149771>] nameidata_to_filp+0x27/0x37
[<c01497b4>] do_filp_open+0x33/0x3b
[<c0149801>] do_sys_open+0x45/0xc9
[<c01498bd>] sys_open+0x1c/0x1e
[<c0102540>] syscall_call+0x7/0xb
```

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

Cc: Paul Menage <menage@google.com>

```
kernel/container.c | 33 ++++++-----
1 file changed, 19 insertions(+), 14 deletions(-)
```

Index: 2.6.22-rc4-mm2/kernel/container.c

=====

--- 2.6.22-rc4-mm2.orig/kernel/container.c

+++ 2.6.22-rc4-mm2/kernel/container.c

@@ -1651,21 +1651,26 @@ static int container_tasks_open(struct i

```

    * show up until sometime later on.
    */
    npids = container_task_count(cont);
-   pidarray = kmalloc(npids * sizeof(pid_t), GFP_KERNEL);
-   if (!pidarray)
-       goto err1;
-
-   npids = pid_array_load(pidarray, npids, cont);
-   sort(pidarray, npids, sizeof(pid_t), cmppid, NULL);
-
-   /* Call pid_array_to_buf() twice, first just to get bufsz */
-   ctr->bufsz = pid_array_to_buf(&c, sizeof(c), pidarray, npids) + 1;
-   ctr->buf = kmalloc(ctr->bufsz, GFP_KERNEL);
-   if (!ctr->buf)
-       goto err2;
-   ctr->bufsz = pid_array_to_buf(ctr->buf, ctr->bufsz, pidarray, npids);
+   if (npids) {
+       pidarray = kmalloc(npids * sizeof(pid_t), GFP_KERNEL);
+       if (!pidarray)
+           goto err1;
+
+       npids = pid_array_load(pidarray, npids, cont);
+       sort(pidarray, npids, sizeof(pid_t), cmppid, NULL);
+
+       /* Call pid_array_to_buf() twice, first just to get bufsz */
+       ctr->bufsz = pid_array_to_buf(&c, sizeof(c), pidarray, npids) + 1;
+       ctr->buf = kmalloc(ctr->bufsz, GFP_KERNEL);
+       if (!ctr->buf)
+           goto err2;
+       ctr->bufsz = pid_array_to_buf(ctr->buf, ctr->bufsz, pidarray, npids);
-   kfree(pidarray);
+   kfree(pidarray);
+   } else {
+       ctr->buf = 0;
+       ctr->bufsz = 0;
+   }
    file->private_data = ctr;
    return 0;

```

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
