

---

Subject: [PATCH 10/10] Task Containers(V11): Simple task container debug info subsystem

Posted by [Paul Menage](#) on Fri, 20 Jul 2007 18:32:02 GMT

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

---

This example subsystem exports debugging information as an aid to diagnosing refcount leaks, etc, in the container framework.

Signed-off-by: Paul Menage <menage@google.com>

---

```
include/linux/container_subsys.h | 4 +
init/Kconfig                     | 10 ++++
kernel/Makefile                  | 1
kernel/container_debug.c         | 97 +++++
4 files changed, 112 insertions(+)
```

Index: container-2.6.22-rc6-mm1/include/linux/container\_subsys.h

```
=====
--- container-2.6.22-rc6-mm1.orig/include/linux/container_subsys.h
+++ container-2.6.22-rc6-mm1/include/linux/container_subsys.h
@@ -19,4 +19,8 @@ SUBSYS(cpuacct)
```

```
/* */
```

```
+#ifdef CONFIG_CONTAINER_DEBUG
+SUBSYS(debug)
+#endif
+
/* */
```

Index: container-2.6.22-rc6-mm1/init/Kconfig

```
=====
--- container-2.6.22-rc6-mm1.orig/init/Kconfig
+++ container-2.6.22-rc6-mm1/init/Kconfig
@@ -303,6 +303,16 @@ config CONTAINERS
```

Say N if unsure.

```
+config CONTAINER_DEBUG
+ bool "Example debug container subsystem"
+ depends on CONTAINERS
+ help
+   This option enables a simple container subsystem that
+   exports useful debugging information about the containers
+   framework
+
+ Say N if unsure
+
```

config CPUSETS

bool "Cpuset support"

depends on SMP && CONTAINERS

Index: container-2.6.22-rc6-mm1/kernel/Makefile

=====  
--- container-2.6.22-rc6-mm1.orig/kernel/Makefile

+++ container-2.6.22-rc6-mm1/kernel/Makefile

@ @ -38,6 +38,7 @ @ obj-\$(CONFIG\_BSD\_PROCESS\_ACCT) += acct.o

obj-\$(CONFIG\_KEXEC) += kexec.o

obj-\$(CONFIG\_COMPAT) += compat.o

obj-\$(CONFIG\_CONTAINERS) += container.o

+obj-\$(CONFIG\_CONTAINER\_DEBUG) += container\_debug.o

obj-\$(CONFIG\_CPUSETS) += cpuset.o

obj-\$(CONFIG\_CONTAINER\_CPUACCT) += cpu\_acct.o

obj-\$(CONFIG\_IKCONFIG) += configs.o

Index: container-2.6.22-rc6-mm1/kernel/container\_debug.c

=====  
--- /dev/null

+++ container-2.6.22-rc6-mm1/kernel/container\_debug.c

@ @ -0,0 +1,97 @ @

+/\*

+ \* kernel/ccontainer\_debug.c - Example container subsystem that

+ \* exposes debug info

+ \*

+ \* Copyright (C) Google Inc, 2007

+ \*

+ \* Developed by Paul Menage (menage@google.com)

+ \*

+ \*/

+

+#include <linux/container.h>

+#include <linux/fs.h>

+#include <linux/slab.h>

+#include <linux/rcupdate.h>

+

+#include <asm/atomic.h>

+

+static struct container\_subsys\_state \*debug\_create(struct container\_subsys \*ss,

+ struct container \*cont)

+{

+ struct container\_subsys\_state \*css = kzalloc(sizeof(\*css), GFP\_KERNEL);

+

+ if (!css)

+ return ERR\_PTR(-ENOMEM);

+

+ return css;

+}

+

```

+static void debug_destroy(struct container_subsys *ss, struct container *cont)
+{
+ kfree(cont->subsys[debug_subsys_id]);
+}
+
+static u64 container_refcount_read(struct container *cont, struct cftype *cft)
+{
+ return atomic_read(&cont->count);
+}
+
+static u64 taskcount_read(struct container *cont, struct cftype *cft)
+{
+ u64 count;
+
+ container_lock();
+ count = container_task_count(cont);
+ container_unlock();
+ return count;
+}
+
+static u64 current_css_group_read(struct container *cont, struct cftype *cft)
+{
+ return (u64)(long)current->containers;
+}
+
+static u64 current_css_group_refcount_read(struct container *cont,
+      struct cftype *cft)
+{
+ u64 count;
+
+ rcu_read_lock();
+ count = atomic_read(&current->containers->ref.refcount);
+ rcu_read_unlock();
+ return count;
+}
+
+static struct cftype files[] = {
+ {
+ .name = "container_refcount",
+ .read_uint = container_refcount_read,
+ },
+ {
+ .name = "taskcount",
+ .read_uint = taskcount_read,
+ },
+ {
+ .name = "current_css_group",

```

```
+ .read_uint = current_css_group_read,  
+ },  
+  
+ {  
+ .name = "current_css_group_refcount",  
+ .read_uint = current_css_group_refcount_read,  
+ },  
+};  
+  
+static int debug_populate(struct container_subsys *ss, struct container *cont)  
+{  
+ return container_add_files(cont, ss, files, ARRAY_SIZE(files));  
+}  
+  
+struct container_subsys debug_subsys = {  
+ .name = "debug",  
+ .create = debug_create,  
+ .destroy = debug_destroy,  
+ .populate = debug_populate,  
+ .subsys_id = debug_subsys_id,  
+};  
  
--
```

---