Subject: [PATCH] Add cgroup write_uint() helper method
Posted by Paul Menage on Wed, 26 Sep 2007 03:12:22 GMT
View Forum Message <> Reply to Message

Add write_uint() helper method for cgroup subsystems

This helper is analagous to the read_uint() helper method for
reporting u64 values to userspace. It's designed to reduce the amount
of boilerplate requierd for creating new cgroup subsystems.

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

```
---
 include/linux/cgroup.h |    8 ++++++++
 kernel/cgroup.c        |   42 ++++++++++++++++++++++++++++++++++++++++----
 2 files changed, 46 insertions(+), 4 deletions(-)

Index: container-2.6.23-rc8-mm1/include/linux/cgroup.h
===================================================================
--- container-2.6.23-rc8-mm1.orig/include/linux/cgroup.h
+++ container-2.6.23-rc8-mm1/include/linux/cgroup.h
@@ -196,6 +196,14 @@ struct cftype {
  ssize_t (*write) (struct cgroup *cont, struct cftype *cft,
      struct file *file,
      const char __user *buf, size_t nbytes, loff_t *ppos);
+
+ /*
+  * write_uint() is a shortcut for the common case of accepting
+  * a single integer (as parsed by simple_strtoull) from
+  * userspace. Use in place of write(); return 0 or error.
+  */
+ int (*write_uint) (struct cgroup *cont, struct cftype *cft, u64 val);
+
  int (*release) (struct inode *inode, struct file *file);
 };

Index: container-2.6.23-rc8-mm1/kernel/cgroup.c
===================================================================
--- container-2.6.23-rc8-mm1.orig/kernel/cgroup.c
+++ container-2.6.23-rc8-mm1/kernel/cgroup.c
@@ -1274,6 +1274,39 @@ enum cgroup_filetype {
  FILE_RELEASE_AGENT,
 };

+static ssize_t cgroup_write_uint(struct cgroup *cont, struct cftype *cft,
+     struct file *file,
+     const char __user *userbuf,
+     size_t nbytes, loff_t *unused_ppos)
```

```
+{
+ char buffer[64];
+ int retval = 0;
+ u64 val;
+ char *end;
+
+ if (!nbytes)
+  return -EINVAL;
+ if (nbytes >= sizeof(buffer))
+  return -E2BIG;
+ if (copy_from_user(buffer, userbuf, nbytes))
+  return -EFAULT;
+
+ buffer[nbytes] = 0;     /* nul-terminate */
+
+ /* strip newline if necessary */
+ if (nbytes && (buffer[nbytes-1] == '\n'))
+  buffer[nbytes-1] = 0;
+ val = simple_strtoull(buffer, &end, 0);
+ if (*end)
+  return -EINVAL;
+
+ /* Pass to subsystem */
+ retval = cft->write_uint(cont, cft, val);
+ if (!retval)
+  retval = nbytes;
+ return retval;
+}
+
 static ssize_t cgroup_common_file_write(struct cgroup *cont,
         struct cftype *cft,
         struct file *file,
@@ -1357,10 +1390,11 @@ static ssize_t cgroup_file_write(struct

  if (!cft)
   return -ENODEV;
- if (!cft->write)
-  return -EINVAL;
-
- return cft->write(cont, cft, file, buf, nbytes, ppos);
+ if (cft->write)
+  return cft->write(cont, cft, file, buf, nbytes, ppos);
+ if (cft->write_uint)
+  return cgroup_write_uint(cont, cft, file, buf, nbytes, ppos);
+ return -EINVAL;
 }

 static ssize_t cgroup_read_uint(struct cgroup *cont, struct cftype *cft,
```

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