Subject: [PATCH 01/10] CGroup API files: Rename read/write_uint methods to read_write_u64
Posted by Paul Menage on Sat, 23 Feb 2008 22:47:26 GMT
View Forum Message <> Reply to Message

Several people have justifiably complained that the "_uint" suffix is
inappropriate for functions that handle u64 values, so this patch just
renames all these functions and their users to have the suffic _u64.

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

```
---
 include/linux/cgroup.h |    8 ++++----
 kernel/cgroup.c        |   32 +++++++++++++++++----------------
 kernel/cgroup_debug.c  |    8 ++++----
 kernel/sched.c         |   18 +++++++++---------
 4 files changed, 33 insertions(+), 33 deletions(-)

Index: cgroup-2.6.25-rc2-mm1/include/linux/cgroup.h
===================================================================
--- cgroup-2.6.25-rc2-mm1.orig/include/linux/cgroup.h
+++ cgroup-2.6.25-rc2-mm1/include/linux/cgroup.h
@@ -190,20 +190,20 @@ struct cftype {
     struct file *file,
     char __user *buf, size_t nbytes, loff_t *ppos);
  /*
-  * read_uint() is a shortcut for the common case of returning a
+  * read_u64() is a shortcut for the common case of returning a
   * single integer. Use it in place of read()
   */
- u64 (*read_uint) (struct cgroup *cont, struct cftype *cft);
+ u64 (*read_u64) (struct cgroup *cont, struct cftype *cft);
  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
+  * write_u64() 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 (*write_u64) (struct cgroup *cont, struct cftype *cft, u64 val);

  int (*release) (struct inode *inode, struct file *file);
 };
Index: cgroup-2.6.25-rc2-mm1/kernel/cgroup.c
```

```
=====================================================================
--- cgroup-2.6.25-rc2-mm1.orig/kernel/cgroup.c
+++ cgroup-2.6.25-rc2-mm1/kernel/cgroup.c
@@ -1303,10 +1303,10 @@ enum cgroup_filetype {
  FILE_RELEASE_AGENT,
 };

-static ssize_t cgroup_write_uint(struct cgroup *cgrp, struct cftype *cft,
-     struct file *file,
-     const char __user *userbuf,
-     size_t nbytes, loff_t *unused_ppos)
+static ssize_t cgroup_write_u64(struct cgroup *cgrp, struct cftype *cft,
+    struct file *file,
+    const char __user *userbuf,
+    size_t nbytes, loff_t *unused_ppos)
 {
  char buffer[64];
  int retval = 0;
@@ -1330,7 +1330,7 @@ static ssize_t cgroup_write_uint(struct
   return -EINVAL;

  /* Pass to subsystem */
- retval = cft->write_uint(cgrp, cft, val);
+ retval = cft->write_u64(cgrp, cft, val);
  if (!retval)
   retval = nbytes;
  return retval;
@@ -1411,18 +1411,18 @@ static ssize_t cgroup_file_write(struct
   return -ENODEV;
  if (cft->write)
   return cft->write(cgrp, cft, file, buf, nbytes, ppos);
- if (cft->write_uint)
-  return cgroup_write_uint(cgrp, cft, file, buf, nbytes, ppos);
+ if (cft->write_u64)
+  return cgroup_write_u64(cgrp, cft, file, buf, nbytes, ppos);
  return -EINVAL;
 }

-static ssize_t cgroup_read_uint(struct cgroup *cgrp, struct cftype *cft,
-       struct file *file,
-       char __user *buf, size_t nbytes,
-       loff_t *ppos)
+static ssize_t cgroup_read_u64(struct cgroup *cgrp, struct cftype *cft,
+         struct file *file,
+         char __user *buf, size_t nbytes,
+         loff_t *ppos)
 {
  char tmp[64];
```

```
- u64 val = cft->read_uint(cgrp, cft);
+ u64 val = cft->read_u64(cgrp, cft);
  int len = sprintf(tmp, "%llu\n", (unsigned long long) val);

  return simple_read_from_buffer(buf, nbytes, ppos, tmp, len);
@@ -1482,8 +1482,8 @@ static ssize_t cgroup_file_read(struct f

  if (cft->read)
   return cft->read(cgrp, cft, file, buf, nbytes, ppos);
- if (cft->read_uint)
-  return cgroup_read_uint(cgrp, cft, file, buf, nbytes, ppos);
+ if (cft->read_u64)
+  return cgroup_read_u64(cgrp, cft, file, buf, nbytes, ppos);
  return -EINVAL;
 }

@@ -2141,14 +2141,14 @@ static struct cftype files[] = {

  {
   .name = "notify_on_release",
-  .read_uint = cgroup_read_notify_on_release,
+  .read_u64 = cgroup_read_notify_on_release,
   .write = cgroup_common_file_write,
   .private = FILE_NOTIFY_ON_RELEASE,
  },

  {
   .name = "releasable",
-  .read_uint = cgroup_read_releasable,
+  .read_u64 = cgroup_read_releasable,
   .private = FILE_RELEASABLE,
  }
 };
Index: cgroup-2.6.25-rc2-mm1/kernel/cgroup_debug.c
===================================================================
--- cgroup-2.6.25-rc2-mm1.orig/kernel/cgroup_debug.c
+++ cgroup-2.6.25-rc2-mm1/kernel/cgroup_debug.c
@@ -65,21 +65,21 @@ static u64 current_css_set_refcount_read
 static struct cftype files[] =  {
  {
   .name = "cgroup_refcount",
-  .read_uint = cgroup_refcount_read,
+  .read_u64 = cgroup_refcount_read,
  },
  {
   .name = "taskcount",
-  .read_uint = taskcount_read,
+  .read_u64 = taskcount_read,
```

```
 },

 {
  .name = "current_css_set",
-  .read_uint = current_css_set_read,
+  .read_u64 = current_css_set_read,
 },

 {
  .name = "current_css_set_refcount",
-  .read_uint = current_css_set_refcount_read,
+  .read_u64 = current_css_set_refcount_read,
 },
};

Index: cgroup-2.6.25-rc2-mm1/kernel/sched.c
===================================================================
--- cgroup-2.6.25-rc2-mm1.orig/kernel/sched.c
+++ cgroup-2.6.25-rc2-mm1/kernel/sched.c
@@ -8286,13 +8286,13 @@ cpu_cgroup_attach(struct cgroup_subsys *
}

#ifdef CONFIG_FAIR_GROUP_SCHED
-static int cpu_shares_write_uint(struct cgroup *cgrp, struct cftype *cftype,
+static int cpu_shares_write_u64(struct cgroup *cgrp, struct cftype *cftype,
    u64 shareval)
{
 return sched_group_set_shares(cgroup_tg(cgrp), shareval);
}

-static u64 cpu_shares_read_uint(struct cgroup *cgrp, struct cftype *cft)
+static u64 cpu_shares_read_u64(struct cgroup *cgrp, struct cftype *cft)
{
 struct task_group *tg = cgroup_tg(cgrp);

@@ -8346,13 +8346,13 @@ static ssize_t cpu_rt_runtime_read(struc
 return simple_read_from_buffer(buf, nbytes, ppos, tmp, len);
}

-static int cpu_rt_period_write_uint(struct cgroup *cgrp, struct cftype *cftype,
+static int cpu_rt_period_write_u64(struct cgroup *cgrp, struct cftype *cftype,
   u64 rt_period_us)
{
 return sched_group_set_rt_period(cgroup_tg(cgrp), rt_period_us);
}

-static u64 cpu_rt_period_read_uint(struct cgroup *cgrp, struct cftype *cft)
+static u64 cpu_rt_period_read_u64(struct cgroup *cgrp, struct cftype *cft)
```

```
 {
  return sched_group_rt_period(cgroup_tg(cgrp));
 }
@@ -8362,8 +8362,8 @@ static struct cftype cpu_files[] = {
 #ifdef CONFIG_FAIR_GROUP_SCHED
  {
   .name = "shares",
-  .read_uint = cpu_shares_read_uint,
-  .write_uint = cpu_shares_write_uint,
+  .read_u64 = cpu_shares_read_u64,
+  .write_u64 = cpu_shares_write_u64,
  },
 #endif
 #ifdef CONFIG_RT_GROUP_SCHED
@@ -8374,8 +8374,8 @@ static struct cftype cpu_files[] = {
  },
  {
   .name = "rt_period_us",
-  .read_uint = cpu_rt_period_read_uint,
-  .write_uint = cpu_rt_period_write_uint,
+  .read_u64 = cpu_rt_period_read_u64,
+  .write_u64 = cpu_rt_period_write_u64,
  },
 #endif
 };
@@ -8483,7 +8483,7 @@ static u64 cpuusage_read(struct cgroup *
 static struct cftype files[] = {
  {
   .name = "usage",
-  .read_uint = cpuusage_read,
+  .read_u64 = cpuusage_read,
  },
 };
```

--

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