

---

Subject: [PATCH 3/3] proc: make UTS-related sysctls utsns aware  
Posted by [Sam Vilain](#) on Tue, 23 May 2006 01:23:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: Sam Vilain <sam.vilain@catalyst.net.nz>

Add a new function proc\_do\_utsns\_string, that derives the pointer into the uts\_namespace->name structure, currently based on the filename of the dentry in /proc, and calls \_proc\_do\_string()

---

RFC only - not tested yet. builds, though.

kernel/sysctl.c | 104 +++  
1 files changed, 104 insertions(+), 0 deletions(-)

diff --git a/kernel/sysctl.c b/kernel/sysctl.c

index cf053fc..37dc17f 100644

--- a/kernel/sysctl.c

+++ b/kernel/sysctl.c

@@ -141,8 +141,13 @@ #endif

static int parse\_table(int \_\_user \*, int, void \_\_user \*, size\_t \_\_user \*, void \_\_user \*, size\_t, ctl\_table \*, void \*\*);

+#ifndef CONFIG\_UTS\_NS

static int proc\_do\_uts\_string(ctl\_table \*table, int write, struct file \*filp, void \_\_user \*buffer, size\_t \*lenp, loff\_t \*ppos);

+#else

+static int proc\_do\_utsns\_string(ctl\_table \*table, int write, struct file \*filp,

+ void \_\_user \*buffer, size\_t \*lenp, loff\_t \*ppos);

+#endif

static ctl\_table root\_table[];

static struct ctl\_table\_header root\_table\_header =

@@ -238,6 +243,7 @@ #endif

};

static ctl\_table kern\_table[] = {

+#ifndef CONFIG\_UTS\_NS

{

.ctl\_name = KERN\_OSTYPE,

.procname = "ostype",

@@ -283,6 +289,54 @@ static ctl\_table kern\_table[] = {

.proc\_handler = &proc\_do\_uts\_string,

.strategy = &sysctl\_string,

},

+#else /\* !CONFIG\_UTS\_NS \*/

+ {

+ .ctl\_name = KERN\_OSTYPE,

```

+ .procname = "ostype",
+ .data = NULL,
+ /* could maybe use __NEW_UTS_LEN here? */
+ .maxlen = sizeof(current->nsproxy->uts_ns->name.sysname),
+ .mode = 0444,
+ .proc_handler = &proc_do_utsns_string,
+ .strategy = &sysctl_string,
+ },
+ {
+ .ctl_name = KERN_OSRELEASE,
+ .procname = "osrelease",
+ .data = NULL,
+ .maxlen = sizeof(current->nsproxy->uts_ns->name.release),
+ .mode = 0444,
+ .proc_handler = &proc_do_utsns_string,
+ .strategy = &sysctl_string,
+ },
+ {
+ .ctl_name = KERN_VERSION,
+ .procname = "version",
+ .data = NULL,
+ .maxlen = sizeof(current->nsproxy->uts_ns->name.version),
+ .mode = 0444,
+ .proc_handler = &proc_do_utsns_string,
+ .strategy = &sysctl_string,
+ },
+ {
+ .ctl_name = KERN_NODENAME,
+ .procname = "hostname",
+ .data = NULL,
+ .maxlen = sizeof(current->nsproxy->uts_ns->name.nodename),
+ .mode = 0644,
+ .proc_handler = &proc_do_utsns_string,
+ .strategy = &sysctl_string,
+ },
+ {
+ .ctl_name = KERN_DOMAINNAME,
+ .procname = "domainname",
+ .data = NULL,
+ .maxlen = sizeof(current->nsproxy->uts_ns->name.domainname),
+ .mode = 0644,
+ .proc_handler = &proc_do_utsns_string,
+ .strategy = &sysctl_string,
+ },
+ #endif /* !CONFIG_UTS_NS */
+ {
+ .ctl_name = KERN_PANIC,
+ .procname = "panic",

```

```

@@ -1684,6 +1738,7 @@ int proc_dostring(ctl_table *table, int
 * to observe. Should this be in kernel/sys.c ???
 */

+#ifndef CONFIG_UTS_NS
static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
void __user *buffer, size_t *lenp, loff_t *ppos)
{
@@ -1700,6 +1755,55 @@ static int proc_do_uts_string(ctl_table
}
return r;
}
#else /* !CONFIG_UTS_NS */
+static int proc_do_utsns_string(ctl_table *table, int write, struct file *filp,
+ void __user *buffer, size_t *lenp, loff_t *ppos)
+{
+ int r;
+ struct uts_namespace* uts_ns = current->nsproxy->uts_ns;
+ char* which;
+
+ /* map the filename to the pointer. perhaps it would be better
+ to put struct offset pointers in table->data ? */
+ switch (filp->f_dentry->d_name.name[3]) {
+ case 'y': /* ostype */
+ which = uts_ns->name.sysname;
+ break;
+ case 't': /* hostname */
+ which = uts_ns->name.nodename;
+ break;
+ case 'e': /* release */
+ which = uts_ns->name.release;
+ break;
+ case 's': /* version */
+ which = uts_ns->name.version;
+ break;
+ case 'x': /* XXX - unreachable */
+ which = uts_ns->name.machine;
+ break;
+ case 'a': /* domainname */
+ which = uts_ns->name.domainname;
+ break;
+ default:
+ printk("procfs: impossible uts part '%s'",
+ (char*)filp->f_dentry->d_name.name);
+ r = -EINVAL;
+ goto out;
+ }
+
+

```

```
+ if (!write) {
+   down_read(&uts_sem);
+   r=_proc_do_string(which,table->maxlen,0,filp,buffer,lenp, ppos);
+   up_read(&uts_sem);
+ } else {
+   down_write(&uts_sem);
+   r=_proc_do_string(which,table->maxlen,1,filp,buffer,lenp, ppos);
+   up_write(&uts_sem);
+ }
+ out:
+ return r;
+}
+#endif /* !CONFIG_UTS_NS */
```

```
static int do_proc_dointvec_conv(int *negp, unsigned long *lvalp,
    int *valp,
```

---