
Subject: Re: [PATCH 3/3] proc: make UTS-related sysctls utsns aware
Posted by [Sam Vilain](#) on Thu, 25 May 2006 04:01:44 GMT

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

Dave Hansen wrote:

> On Tue, 2006-05-23 at 13:23 +1200, Sam Vilain wrote:

```
>
> >> + /* 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;
>
>
> Why not just switch on the ->ctl_name from the table? Wouldn't that be
> easier?
```

Ha! Sure would.

Subject: proc: make UTS-related sysctls utsns aware
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()

Signed-off-by: Sam Vilain <sam.vilain@catalyst.net.nz>

Cc: Serge E. Hallyn <serue@us.ibm.com>

Cc: Kirill Korotaev <dev@openvz.org>

Cc: "Eric W. Biederman" <ebiederm@xmission.com>

Cc: Herbert Poetzl <herbert@13thfloor.at>

Cc: Andrey Savochkin <saw@sw.ru>

Cc: Andrew Morton <akpm@osdl.org>

kernel/sysctl.c | 97 ++++++
1 files changed, 97 insertions(+), 0 deletions(-)

```
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index cf053fc..5e6837e 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,
},
+/* !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,
+ },
+#
+/* !CONFIG_UTS_NS */
{
.ctl_name = KERN_PANIC,
.procname = "panic",
@@ -1684,6 +1738,7 @@ int proc_destring(ctl_table *table, int
* to observe. Should this be in kernel/sys.c ?????
*/
#endif /* !CONFIG_UTS_NS */

#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 +1755,48 @@ 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;
+
+ switch (table->ctl_name) {
+ case KERN_OSTYPE:
+ which = uts_ns->name.sysname;

```

```

+ break;
+ case KERN_NODENAME:
+ which = uts_ns->name.nodename;
+ break;
+ case KERN_OSRELEASE:
+ which = uts_ns->name.release;
+ break;
+ case KERN_VERSION:
+ which = uts_ns->name.version;
+ break;
+ case KERN_DOMAINNAME:
+ which = uts_ns->name.domainname;
+ break;
+ default:
+ 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,

```
