Subject: [PATCH 51/59] sysctl: Move SYSV IPC sysctls to their own file
Posted by ebiederm on Tue, 16 Jan 2007 16:39:56 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This is just a simple cleanup to keep kernel/sysctl.c
from getting to crowded with special cases, and by
keeping all of the ipc logic to together it makes
the code a little more readable.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 init/Kconfig | 6 ++
 ipc/Makefile | 1 +
 ipc/ipc_sysctl.c | 182 ++
 kernel/sysctl.c | 174 --
 4 files changed, 189 insertions(+), 174 deletions(-)

diff --git a/init/Kconfig b/init/Kconfig
index a3f83e2..33bc38d 100644
--- a/init/Kconfig
+++ b/init/Kconfig
@@ -116,6 +116,12 @@ config SYSVIPC
    section 6.4 of the Linux Programmer's Guide, available from
    <http://www.tldp.org/guides.html>.

+config SYSVIPC_SYSCTL
+ bool
+ depends on SYSVIPC
+ depends on SYSCTL
+ default y
+
 config IPC_NS
  bool "IPC Namespaces"
  depends on SYSVIPC
diff --git a/ipc/Makefile b/ipc/Makefile
index 0a6d626..b93bba6 100644
--- a/ipc/Makefile
+++ b/ipc/Makefile
@@ -4,6 +4,7 @@

 obj-$(CONFIG_SYSVIPC_COMPAT) += compat.o
 obj-$(CONFIG_SYSVIPC) += util.o msgutil.o msg.o sem.o shm.o
+obj-$(CONFIG_SYSVIPC_SYSCTL) += ipc_sysctl.o
 obj_mq-$(CONFIG_COMPAT) += compat_mq.o
 obj-$(CONFIG_POSIX_MQUEUE) += mqueue.o msgutil.o $(obj_mq-y)

```
diff --git a/ipc/ipc_sysctl.c b/ipc/ipc_sysctl.c
new file mode 100644
index 0000000..9018009
--- /dev/null
+++ b/ipc/ipc_sysctl.c
@@ -0,0 +1,182 @@
+/*
+ *  Copyright (C) 2007
+ *
+ *  Author: Eric Biederman <ebiederm@xmision.com>
+ *
+ *  This program is free software; you can redistribute it and/or
+ *  modify it under the terms of the GNU General Public License as
+ *  published by the Free Software Foundation, version 2 of the
+ *  License.
+ */
+
+#include <linux/module.h>
+#include <linux/ipc.h>
+#include <linux/nsproxy.h>
+#include <linux/sysctl.h>
+
+#ifdef CONFIG_IPC_NS
+static void *get_ipc(ctl_table *table)
+{
+ char *which = table->data;
+ struct ipc_namespace *ipc_ns = current->nsproxy->ipc_ns;
+ which = (which - (char *)&init_ipc_ns) + (char *)ipc_ns;
+ return which;
+}
+#else
+#define get_ipc(T) ((T)->data)
+#endif
+
+#ifdef CONFIG_PROC_FS
+static int proc_ipc_dointvec(ctl_table *table, int write, struct file *filp,
+ void __user *buffer, size_t *lenp, loff_t *ppos)
+{
+ struct ctl_table ipc_table;
+ memcpy(&ipc_table, table, sizeof(ipc_table));
+ ipc_table.data = get_ipc(table);
+
+ return proc_dointvec(&ipc_table, write, filp, buffer, lenp, ppos);
+}
+
+static int proc_ipc_doulongvec_minmax(ctl_table *table, int write,
+ struct file *filp, void __user *buffer, size_t *lenp, loff_t *ppos)
+{
```

```
+ struct ctl_table ipc_table;
+ memcpy(&ipc_table, table, sizeof(ipc_table));
+ ipc_table.data = get_ipc(table);
+
+ return proc_doulongvec_minmax(&ipc_table, write, filp, buffer,
+     lenp, ppos);
+}
+
+#else
+#define proc_ipc_do_ulongvec_minmax NULL
+#define proc_ipc_do_intvec     NULL
+#endif
+
+#ifdef CONFIG_SYSCTL_SYSCALL
+/* The generic sysctl ipc data routine. */
+static int sysctl_ipc_data(ctl_table *table, int __user *name, int nlen,
+  void __user *oldval, size_t __user *oldlenp,
+  void __user *newval, size_t newlen)
+{
+ size_t len;
+ void *data;
+
+ /* Get out of I don't have a variable */
+ if (!table->data || !table->maxlen)
+  return -ENOTDIR;
+
+ data = get_ipc(table);
+ if (!data)
+  return -ENOTDIR;
+
+ if (oldval && oldlenp) {
+  if (get_user(len, oldlenp))
+   return -EFAULT;
+  if (len) {
+   if (len > table->maxlen)
+    len = table->maxlen;
+   if (copy_to_user(oldval, data, len))
+    return -EFAULT;
+   if (put_user(len, oldlenp))
+    return -EFAULT;
+  }
+ }
+
+ if (newval && newlen) {
+  if (newlen > table->maxlen)
+   newlen = table->maxlen;
+
+  if (copy_from_user(data, newval, newlen))
```

```
+   return -EFAULT;
+ }
+ return 1;
+}
+#else
+#define sysctl_ipc_data NULL
+#endif
+
+static struct ctl_table ipc_kern_table[] = {
+ {
+  .ctl_name = KERN_SHMMAX,
+  .procname = "shmmax",
+  .data  = &init_ipc_ns.shm_ctlmax,
+  .maxlen  = sizeof (init_ipc_ns.shm_ctlmax),
+  .mode  = 0644,
+  .proc_handler = proc_ipc_doulongvec_minmax,
+  .strategy = sysctl_ipc_data,
+ },
+ {
+  .ctl_name = KERN_SHMALL,
+  .procname = "shmall",
+  .data  = &init_ipc_ns.shm_ctlall,
+  .maxlen  = sizeof (init_ipc_ns.shm_ctlall),
+  .mode  = 0644,
+  .proc_handler = proc_ipc_doulongvec_minmax,
+  .strategy = sysctl_ipc_data,
+ },
+ {
+  .ctl_name = KERN_SHMMNI,
+  .procname = "shmmni",
+  .data  = &init_ipc_ns.shm_ctlmni,
+  .maxlen  = sizeof (init_ipc_ns.shm_ctlmni),
+  .mode  = 0644,
+  .proc_handler = proc_ipc_dointvec,
+  .strategy = sysctl_ipc_data,
+ },
+ {
+  .ctl_name = KERN_MSGMAX,
+  .procname = "msgmax",
+  .data  = &init_ipc_ns.msg_ctlmax,
+  .maxlen  = sizeof (init_ipc_ns.msg_ctlmax),
+  .mode  = 0644,
+  .proc_handler = proc_ipc_dointvec,
+  .strategy = sysctl_ipc_data,
+ },
+ {
+  .ctl_name = KERN_MSGMNI,
+  .procname = "msgmni",
```

```
+  .data  = &init_ipc_ns.msg_ctlmni,
+  .maxlen  = sizeof (init_ipc_ns.msg_ctlmni),
+  .mode  = 0644,
+  .proc_handler = proc_ipc_dointvec,
+  .strategy = sysctl_ipc_data,
+ },
+ {
+  .ctl_name = KERN_MSGMNB,
+  .procname =  "msgmnb",
+  .data  = &init_ipc_ns.msg_ctlmnb,
+  .maxlen  = sizeof (init_ipc_ns.msg_ctlmnb),
+  .mode  = 0644,
+  .proc_handler = proc_ipc_dointvec,
+  .strategy = sysctl_ipc_data,
+ },
+ {
+  .ctl_name = KERN_SEM,
+  .procname = "sem",
+  .data  = &init_ipc_ns.sem_ctls,
+  .maxlen  = 4*sizeof (int),
+  .mode  = 0644,
+  .proc_handler = proc_ipc_dointvec,
+  .strategy = sysctl_ipc_data,
+ },
+ {}
+};
+
+static struct ctl_table ipc_root_table[] = {
+ {
+  .ctl_name = CTL_KERN,
+  .procname = "kernel",
+  .mode  = 0555,
+  .child  = ipc_kern_table,
+ },
+ {}
+};
+
+static int __init ipc_sysctl_init(void)
+{
+ register_sysctl_table(ipc_root_table, 0);
+ return 0;
+}
+
+__initcall(ipc_sysctl_init);
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index a8c0a03..6e2e608 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
```

```
@@ -90,12 +90,6 @@ extern char modprobe_path[];
 #ifdef CONFIG_CHR_DEV_SG
 extern int sg_big_buff;
 #endif
-#ifdef CONFIG_SYSVIPC
-static int proc_ipc_dointvec(ctl_table *table, int write, struct file *filp,
-  void __user *buffer, size_t *lenp, loff_t *ppos);
-static int proc_ipc_doulongvec_minmax(ctl_table *table, int write, struct file *filp,
-  void __user *buffer, size_t *lenp, loff_t *ppos);
-#endif

 #ifdef __sparc__
 extern char reboot_command [];
@@ -135,11 +129,6 @@ static int parse_table(int __user *, int, void __user *, size_t __user *,
   void __user *, size_t, ctl_table *);
 #endif

-#ifdef CONFIG_SYSVIPC
-static int sysctl_ipc_data(ctl_table *table, int __user *name, int nlen,
-    void __user *oldval, size_t __user *oldlenp,
-    void __user *newval, size_t newlen);
-#endif

 #ifdef CONFIG_PROC_SYSCTL
 static int proc_do_cad_pid(ctl_table *table, int write, struct file *filp,
@@ -168,17 +157,6 @@ int sysctl_legacy_va_layout;
 #endif


-#ifdef CONFIG_SYSVIPC
-static void *get_ipc(ctl_table *table, int write)
-{
- char *which = table->data;
- struct ipc_namespace *ipc_ns = current->nsproxy->ipc_ns;
- which = (which - (char *)&init_ipc_ns) + (char *)ipc_ns;
- return which;
-}
-#else
-#define get_ipc(T,W) ((T)->data)
-#endif

 /* /proc declarations: */

@@ -400,71 +378,6 @@ static ctl_table kern_table[] = {
   .proc_handler = &proc_dointvec,
  },
 #endif
-#ifdef CONFIG_SYSVIPC
```

```
- {
-  .ctl_name = KERN_SHMMAX,
-  .procname = "shmmax",
-  .data  = &init_ipc_ns.shm_ctlmax,
-  .maxlen  = sizeof (init_ipc_ns.shm_ctlmax),
-  .mode  = 0644,
-  .proc_handler = &proc_ipc_doulongvec_minmax,
-  .strategy = sysctl_ipc_data,
- },
- {
-  .ctl_name = KERN_SHMALL,
-  .procname = "shmall",
-  .data  = &init_ipc_ns.shm_ctlall,
-  .maxlen  = sizeof (init_ipc_ns.shm_ctlall),
-  .mode  = 0644,
-  .proc_handler = &proc_ipc_doulongvec_minmax,
-  .strategy = sysctl_ipc_data,
- },
- {
-  .ctl_name = KERN_SHMMNI,
-  .procname = "shmmni",
-  .data  = &init_ipc_ns.shm_ctlmni,
-  .maxlen  = sizeof (init_ipc_ns.shm_ctlmni),
-  .mode  = 0644,
-  .proc_handler = &proc_ipc_dointvec,
-  .strategy = sysctl_ipc_data,
- },
- {
-  .ctl_name = KERN_MSGMAX,
-  .procname = "msgmax",
-  .data  = &init_ipc_ns.msg_ctlmax,
-  .maxlen  = sizeof (init_ipc_ns.msg_ctlmax),
-  .mode  = 0644,
-  .proc_handler = &proc_ipc_dointvec,
-  .strategy = sysctl_ipc_data,
- },
- {
-  .ctl_name = KERN_MSGMNI,
-  .procname = "msgmni",
-  .data  = &init_ipc_ns.msg_ctlmni,
-  .maxlen  = sizeof (init_ipc_ns.msg_ctlmni),
-  .mode  = 0644,
-  .proc_handler = &proc_ipc_dointvec,
-  .strategy = sysctl_ipc_data,
- },
- {
-  .ctl_name = KERN_MSGMNB,
-  .procname =  "msgmnb",
```

```
-  .data  = &init_ipc_ns.msg_ctlmnb,
-  .maxlen  = sizeof (init_ipc_ns.msg_ctlmnb),
-  .mode  = 0644,
-  .proc_handler = &proc_ipc_dointvec,
-  .strategy = sysctl_ipc_data,
- },
- {
-  .ctl_name = KERN_SEM,
-  .procname = "sem",
-  .data  = &init_ipc_ns.sem_ctls,
-  .maxlen  = 4*sizeof (int),
-  .mode  = 0644,
-  .proc_handler = &proc_ipc_dointvec,
-  .strategy = sysctl_ipc_data,
- },
-#endif
 #ifdef CONFIG_MAGIC_SYSRQ
  {
   .ctl_name = KERN_SYSRQ,
@@ -2240,27 +2153,6 @@ int proc_dointvec_ms_jiffies(ctl_table *table, int write, struct file *filp,
     do_proc_dointvec_ms_jiffies_conv, NULL);
 }

-#ifdef CONFIG_SYSVIPC
-static int proc_ipc_dointvec(ctl_table *table, int write, struct file *filp,
- void __user *buffer, size_t *lenp, loff_t *ppos)
-{
- void *which;
- which = get_ipc(table, write);
- return __do_proc_dointvec(which, table, write, filp, buffer,
-   lenp, ppos, NULL, NULL);
-}
-
-static int proc_ipc_doulongvec_minmax(ctl_table *table, int write,
- struct file *filp, void __user *buffer, size_t *lenp, loff_t *ppos)
-{
- void *which;
- which = get_ipc(table, write);
- return __do_proc_doulongvec_minmax(which, table, write, filp, buffer,
-   lenp, ppos, 1l, 1l);
-}
-
-#endif
-
 static int proc_do_cad_pid(ctl_table *table, int write, struct file *filp,
       void __user *buffer, size_t *lenp, loff_t *ppos)
 {
@@ -2291,25 +2183,6 @@ int proc_dostring(ctl_table *table, int write, struct file *filp,
```

```
  return -ENOSYS;
 }

-#ifdef CONFIG_SYSVIPC
-static int proc_do_ipc_string(ctl_table *table, int write, struct file *filp,
-  void __user *buffer, size_t *lenp, loff_t *ppos)
-{
- return -ENOSYS;
-}
-static int proc_ipc_dointvec(ctl_table *table, int write, struct file *filp,
-  void __user *buffer, size_t *lenp, loff_t *ppos)
-{
- return -ENOSYS;
-}
-static int proc_ipc_doulongvec_minmax(ctl_table *table, int write,
-  struct file *filp, void __user *buffer,
-  size_t *lenp, loff_t *ppos)
-{
- return -ENOSYS;
-}
-#endif
-
 int proc_dointvec(ctl_table *table, int write, struct file *filp,
     void __user *buffer, size_t *lenp, loff_t *ppos)
 {
@@ -2509,47 +2382,6 @@ int sysctl_ms_jiffies(ctl_table *table, int __user *name, int nlen,



-#ifdef CONFIG_SYSVIPC
-/* The generic sysctl ipc data routine. */
-static int sysctl_ipc_data(ctl_table *table, int __user *name, int nlen,
-  void __user *oldval, size_t __user *oldlenp,
-  void __user *newval, size_t newlen)
-{
- size_t len;
- void *data;
-
- /* Get out of I don't have a variable */
- if (!table->data || !table->maxlen)
-  return -ENOTDIR;
-
- data = get_ipc(table, 1);
- if (!data)
-  return -ENOTDIR;
-
- if (oldval && oldlenp) {
-  if (get_user(len, oldlenp))
```

```
-   return -EFAULT;
-  if (len) {
-   if (len > table->maxlen)
-    len = table->maxlen;
-   if (copy_to_user(oldval, data, len))
-    return -EFAULT;
-   if (put_user(len, oldlenp))
-    return -EFAULT;
-  }
- }
-
- if (newval && newlen) {
-  if (newlen > table->maxlen)
-   newlen = table->maxlen;
-
-  if (copy_from_user(data, newval, newlen))
-   return -EFAULT;
- }
- return 1;
-}
-#endif
-
 #else /* CONFIG_SYSCTL_SYSCALL */


@@ -2614,12 +2446,6 @@ int sysctl_ms_jiffies(ctl_table *table, int __user *name, int nlen,
  return -ENOSYS;
 }

-static int sysctl_ipc_data(ctl_table *table, int __user *name, int nlen,
-  void __user *oldval, size_t __user *oldlenp,
-  void __user *newval, size_t newlen)
-{
- return -ENOSYS;
-}
 #endif /* CONFIG_SYSCTL_SYSCALL */

 /*
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers