Subject: [PATCH 0/59] Cleanup sysctl
Posted by ebiederm on Tue, 16 Jan 2007 16:33:47 GMT
View Forum Message <> Reply to Message

There has not been much maintenance on sysctl in years, and as a result is
there is a lot to do to allow future interesting work to happen, and being
ambitious I'm trying to do it all at once :)

The patches in this series fall into several general categories.

- Removal of useless attempts to override the standard sysctls

- Registers of sysctl numbers in sysctl.h so someone else does not use
  the magic number and conflict.

- C99 conversions so it becomes possible to change the layout of
  struct ctl_table without breaking everything.

- Removal of useless claims of module ownership, in the proc dir entries

- Removal of sys_sysctl support where people had used conflicting sysctl
  numbers. Trying to break glibc or other applications by changing the
  ABI is not cool. 9 instances of this in the kernel seems a little
  extreme.

- General enhancements when I got the junk I could see out.

Odds are I missed something, most of the cleanups are simply a result of
me working on the sysctl core and glancing at the users and going: What?

Eric

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

Subject: [PATCH 1/59] sysctl x25: Remove unnecessary insert_at_head from register_sysctl_table.
Posted by ebiederm on Tue, 16 Jan 2007 16:39:06 GMT
View Forum Message <> Reply to Message

Since x25 uses unique binary numbers inserting yourself at the
head of the search list for sysctls so you can override already
registered sysctls is pointless.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 net/x25/sysctl_net_x25.c |    2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/x25/sysctl_net_x25.c b/net/x25/sysctl_net_x25.c
index aabda59..94aff67 100644
--- a/net/x25/sysctl_net_x25.c
+++ b/net/x25/sysctl_net_x25.c
@@ -98,7 +98,7 @@ static struct ctl_table x25_root_table[] = {

 void __init x25_register_sysctl(void)
 {
- x25_table_header = register_sysctl_table(x25_root_table, 1);
+ x25_table_header = register_sysctl_table(x25_root_table, 0);
 }

 void x25_unregister_sysctl(void)
--
1.4.4.1.g278f
```

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

Subject: [PATCH 2/59] sysctl: Move CTL_SUNRPC to sysctl.h where it belongs
Posted by ebiederm on Tue, 16 Jan 2007 16:39:07 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 include/linux/sunrpc/debug.h |    1 -
 include/linux/sysctl.h       |    3 ++-
 2 files changed, 2 insertions(+), 2 deletions(-)

diff --git a/include/linux/sunrpc/debug.h b/include/linux/sunrpc/debug.h
index 60fce3c..b7c7307 100644
--- a/include/linux/sunrpc/debug.h
+++ b/include/linux/sunrpc/debug.h
@@ -78,7 +78,6 @@ void  rpc_unregister_sysctl(void);
  * module currently registers its sysctl table dynamically, the sysctl path
  * for module FOO is <CTL_SUNRPC, CTL_FOODEBUG>.
  */
-#define CTL_SUNRPC 7249 /* arbitrary and hopefully unused */

 enum {
```

```
  CTL_RPCDEBUG = 1,
diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 81480e6..54a9cf5 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -69,7 +69,8 @@ enum
  CTL_DEV=7,  /* Devices */
  CTL_BUS=8,  /* Busses */
  CTL_ABI=9,  /* Binary emulation */
- CTL_CPU=10  /* CPU stuff (speed scaling, etc) */
+ CTL_CPU=10,  /* CPU stuff (speed scaling, etc) */
+ CTL_SUNRPC=7249, /* sunrpc debug */
 };

 /* CTL_BUS names: */
--
1.4.4.1.g278f
```

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

Subject: [PATCH 3/59] sysctl: sunrpc Remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:08 GMT
View Forum Message <> Reply to Message

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

Because the sunrpc sysctls don't conflict with any other
sysctls the setting the insert at head flag to register_sysctl
has no semantic meaning.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/sunrpc/sysctl.c | 2 +-
 net/sunrpc/xprtsock.c | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)

```
diff --git a/net/sunrpc/sysctl.c b/net/sunrpc/sysctl.c
index 82b2752..3852689 100644
--- a/net/sunrpc/sysctl.c
+++ b/net/sunrpc/sysctl.c
@@ -36,7 +36,7 @@ void
 rpc_register_sysctl(void)
 {
  if (!sunrpc_table_header) {
```

```
-  sunrpc_table_header = register_sysctl_table(sunrpc_table, 1);
+  sunrpc_table_header = register_sysctl_table(sunrpc_table, 0);
 #ifdef CONFIG_PROC_FS
   if (sunrpc_table[0].de)
    sunrpc_table[0].de->owner = THIS_MODULE;
diff --git a/net/sunrpc/xprtsock.c b/net/sunrpc/xprtsock.c
index 49cabff..98d1af9 100644
--- a/net/sunrpc/xprtsock.c
+++ b/net/sunrpc/xprtsock.c
@@ -1630,7 +1630,7 @@ int init_socket_xprt(void)
 {
 #ifdef RPC_DEBUG
  if (!sunrpc_table_header) {
-  sunrpc_table_header = register_sysctl_table(sunrpc_table, 1);
+  sunrpc_table_header = register_sysctl_table(sunrpc_table, 0);
 #ifdef CONFIG_PROC_FS
   if (sunrpc_table[0].de)
    sunrpc_table[0].de->owner = THIS_MODULE;
--
1.4.4.1.g278f
```

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

Subject: [PATCH 4/59] sysctl: sunrpc Don't unnecessarily set ctl_table->de
Posted by ebiederm on Tue, 16 Jan 2007 16:39:09 GMT
View Forum Message <> Reply to Message

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

We don't need this to prevent module unload races so remove
the unnecessary code.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 net/sunrpc/sysctl.c   |    8 +-------
 net/sunrpc/xprtsock.c |    7 +------
 2 files changed, 2 insertions(+), 13 deletions(-)

diff --git a/net/sunrpc/sysctl.c b/net/sunrpc/sysctl.c
index 3852689..6a82ed2 100644
--- a/net/sunrpc/sysctl.c
+++ b/net/sunrpc/sysctl.c
@@ -35,14 +35,8 @@ static ctl_table  sunrpc_table[];
 void
```

```
 rpc_register_sysctl(void)
 {
- if (!sunrpc_table_header) {
+ if (!sunrpc_table_header)
   sunrpc_table_header = register_sysctl_table(sunrpc_table, 0);
-#ifdef CONFIG_PROC_FS
-  if (sunrpc_table[0].de)
-   sunrpc_table[0].de->owner = THIS_MODULE;
-#endif
- }
-
 }

 void
diff --git a/net/sunrpc/xprtsock.c b/net/sunrpc/xprtsock.c
index 98d1af9..51964cf 100644
--- a/net/sunrpc/xprtsock.c
+++ b/net/sunrpc/xprtsock.c
@@ -1629,13 +1629,8 @@ struct rpc_xprt *xs_setup_tcp(struct sockaddr *addr, size_t addrlen,
struct rpc_
 int init_socket_xprt(void)
 {
 #ifdef RPC_DEBUG
- if (!sunrpc_table_header) {
+ if (!sunrpc_table_header)
   sunrpc_table_header = register_sysctl_table(sunrpc_table, 0);
-#ifdef CONFIG_PROC_FS
-  if (sunrpc_table[0].de)
-   sunrpc_table[0].de->owner = THIS_MODULE;
-#endif
- }
 #endif

  return 0;
--
1.4.4.1.g278f
```

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

Subject: [PATCH 5/59] sysctl: rose remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:10 GMT
View Forum Message <> Reply to Message

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

The sysctl numbers used are unique so setting the insert_at_head
flag serves no semantic purpose.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/rose/sysctl_net_rose.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/rose/sysctl_net_rose.c b/net/rose/sysctl_net_rose.c
index 8548c7c..0190a07 100644
--- a/net/rose/sysctl_net_rose.c
+++ b/net/rose/sysctl_net_rose.c
@@ -160,7 +160,7 @@ static ctl_table rose_root_table[] = {

 void __init rose_register_sysctl(void)
 {
- rose_table_header = register_sysctl_table(rose_root_table, 1);
+ rose_table_header = register_sysctl_table(rose_root_table, 0);
 }

 void rose_unregister_sysctl(void)
--
1.4.4.1.g278f

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

Subject: [PATCH 6/59] sysctl: netrom remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:11 GMT
View Forum Message <> Reply to Message

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

The sysctl numbers used are unique so setting the insert_at_head
flag serves no semantic purpose, so it is just confusing.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/netrom/sysctl_net_netrom.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/netrom/sysctl_net_netrom.c b/net/netrom/sysctl_net_netrom.c
index 6bb8dda..09f4246 100644
--- a/net/netrom/sysctl_net_netrom.c

```
+++ b/net/netrom/sysctl_net_netrom.c
@@ -192,7 +192,7 @@ static ctl_table nr_root_table[] = {

 void __init nr_register_sysctl(void)
 {
- nr_table_header = register_sysctl_table(nr_root_table, 1);
+ nr_table_header = register_sysctl_table(nr_root_table, 0);
 }

 void nr_unregister_sysctl(void)
--
1.4.4.1.g278f
```

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

Subject: [PATCH 7/59] sysctl: llc remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:12 GMT
View Forum Message <> Reply to Message

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

The sysctl numbers used are unique so setting the insert_at_head
flag serves no semantis purpose, and is just confusing.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/llc/sysctl_net_llc.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

```
diff --git a/net/llc/sysctl_net_llc.c b/net/llc/sysctl_net_llc.c
index 45d7dd9..4aab676 100644
--- a/net/llc/sysctl_net_llc.c
+++ b/net/llc/sysctl_net_llc.c
@@ -116,7 +116,7 @@ static struct ctl_table_header *llc_table_header;

 int __init llc_sysctl_init(void)
 {
- llc_table_header = register_sysctl_table(llc_root_table, 1);
+ llc_table_header = register_sysctl_table(llc_root_table, 0);

  return llc_table_header ? 0 : -ENOMEM;
 }
--
1.4.4.1.g278f
```

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

Subject: [PATCH 8/59] sysctl: ipx remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:13 GMT
View Forum Message <> Reply to Message

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

The sysctl numbers used are unique so setting the insert_at_head
flag servers no semantic purpose and is just confusing.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/ipx/sysctl_net_ipx.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/ipx/sysctl_net_ipx.c b/net/ipx/sysctl_net_ipx.c
index fa57473..0442f44 100644
--- a/net/ipx/sysctl_net_ipx.c
+++ b/net/ipx/sysctl_net_ipx.c
@@ -52,7 +52,7 @@ static struct ctl_table_header *ipx_table_header;

 void ipx_register_sysctl(void)
 {
- ipx_table_header = register_sysctl_table(ipx_root_table, 1);
+ ipx_table_header = register_sysctl_table(ipx_root_table, 0);
 }

 void ipx_unregister_sysctl(void)
--
1.4.4.1.g278f

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

Subject: [PATCH 9/59] sysctl: decnet remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:14 GMT
View Forum Message <> Reply to Message

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

The sysctl numbers used are unique so setting the insert_at_head
flag does not succeed in overriding any sysctls, and is just
confusing because it doesn't. Clear the flag.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/decnet/sysctl_net_decnet.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/decnet/sysctl_net_decnet.c b/net/decnet/sysctl_net_decnet.c
index a4065eb..81469fd 100644
--- a/net/decnet/sysctl_net_decnet.c
+++ b/net/decnet/sysctl_net_decnet.c
@@ -491,7 +491,7 @@ static ctl_table dn_root_table[] = {

 void dn_register_sysctl(void)
 {
- dn_table_header = register_sysctl_table(dn_root_table, 1);
+ dn_table_header = register_sysctl_table(dn_root_table, 0);
 }

 void dn_unregister_sysctl(void)
--
1.4.4.1.g278f

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

Subject: [PATCH 10/59] sysctl: dccp remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:15 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/dccp/sysctl.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/dccp/sysctl.c b/net/dccp/sysctl.c
index fdcfca3..3391631 100644
--- a/net/dccp/sysctl.c
+++ b/net/dccp/sysctl.c

```
@@ -127,7 +127,7 @@ static struct ctl_table_header *dccp_table_header;

 int __init dccp_sysctl_init(void)
 {
- dccp_table_header = register_sysctl_table(dccp_root_table, 1);
+ dccp_table_header = register_sysctl_table(dccp_root_table, 0);

  return dccp_table_header != NULL ? 0 : -ENOMEM;
 }
--
1.4.4.1.g278f
```

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

Subject: [PATCH 11/59] sysctl: ax25 remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:16 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/ax25/sysctl_net_ax25.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

```
diff --git a/net/ax25/sysctl_net_ax25.c b/net/ax25/sysctl_net_ax25.c
index d23a27f..afdba04 100644
--- a/net/ax25/sysctl_net_ax25.c
+++ b/net/ax25/sysctl_net_ax25.c
@@ -245,7 +245,7 @@ void ax25_register_sysctl(void)

  ax25_dir_table[0].child = ax25_table;

- ax25_table_header = register_sysctl_table(ax25_root_table, 1);
+ ax25_table_header = register_sysctl_table(ax25_root_table, 0);
 }

 void ax25_unregister_sysctl(void)
--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org

Subject: [PATCH 12/59] sysctl: atalk remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:17 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
net/appletalk/sysctl_net_atalk.c | 2 +-
1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/net/appletalk/sysctl_net_atalk.c b/net/appletalk/sysctl_net_atalk.c
index 40b0af7..4f806b6 100644
--- a/net/appletalk/sysctl_net_atalk.c
+++ b/net/appletalk/sysctl_net_atalk.c
@@ -73,7 +73,7 @@ static struct ctl_table_header *atalk_table_header;

void atalk_register_sysctl(void)
{
- atalk_table_header = register_sysctl_table(atalk_root_table, 1);
+ atalk_table_header = register_sysctl_table(atalk_root_table, 0);
}

void atalk_unregister_sysctl(void)
--
1.4.4.1.g278f

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

Subject: [PATCH 13/59] sysctl: xfs remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:18 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
fs/xfs/linux-2.6/xfs_sysctl.c | 2 +-
1 files changed, 1 insertions(+), 1 deletions(-)

```
diff --git a/fs/xfs/linux-2.6/xfs_sysctl.c b/fs/xfs/linux-2.6/xfs_sysctl.c
index af24653..af777e9 100644
--- a/fs/xfs/linux-2.6/xfs_sysctl.c
+++ b/fs/xfs/linux-2.6/xfs_sysctl.c
@@ -149,7 +149,7 @@ STATIC ctl_table xfs_root_table[] = {
 void
 xfs_sysctl_register(void)
 {
- xfs_table_header = register_sysctl_table(xfs_root_table, 1);
+ xfs_table_header = register_sysctl_table(xfs_root_table, 0);
 }

 void
--
1.4.4.1.g278f
```

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

Subject: [PATCH 14/59] sysctl: C99 convert xfs ctl_tables
Posted by ebiederm on Tue, 16 Jan 2007 16:39:19 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 fs/xfs/linux-2.6/xfs_sysctl.c | 258 ++++++

- &sysctl_intvec, NULL,
- &xfs_params.sgid_inherit.min, &xfs_params.sgid_inherit.max},
-
- {XFS_SYMLINK_MODE, "irix_symlink_mode", &xfs_params.symlink_mode.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.symlink_mode.min, &xfs_params.symlink_mode.max},
-
- {XFS_PANIC_MASK, "panic_mask", &xfs_params.panic_mask.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.panic_mask.min, &xfs_params.panic_mask.max},
-
- {XFS_ERRLEVEL, "error_level", &xfs_params.error_level.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.error_level.min, &xfs_params.error_level.max},
-
- {XFS_SYNCD_TIMER, "xfssyncd_centisecs", &xfs_params.syncd_timer.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.syncd_timer.min, &xfs_params.syncd_timer.max},
-
- {XFS_INHERIT_SYNC, "inherit_sync", &xfs_params.inherit_sync.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.inherit_sync.min, &xfs_params.inherit_sync.max},
-
- {XFS_INHERIT_NODUMP, "inherit_nodump", &xfs_params.inherit_nodump.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.inherit_nodump.min, &xfs_params.inherit_nodump.max},
-
- {XFS_INHERIT_NOATIME, "inherit_noatime", &xfs_params.inherit_noatim.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.inherit_noatim.min, &xfs_params.inherit_noatim.max},
-
- {XFS_BUF_TIMER, "xfsbufd_centisecs", &xfs_params.xfs_buf_timer.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.xfs_buf_timer.min, &xfs_params.xfs_buf_timer.max},
-
- {XFS_BUF_AGE, "age_buffer_centisecs", &xfs_params.xfs_buf_age.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.xfs_buf_age.min, &xfs_params.xfs_buf_age.max},
-

```
- {XFS_INHERIT_NOSYM, "inherit_nosymlinks", &xfs_params.inherit_nosym.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.inherit_nosym.min, &xfs_params.inherit_nosym.max},
-
- {XFS_ROTORSTEP, "rotorstep", &xfs_params.rotorstep.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.rotorstep.min, &xfs_params.rotorstep.max},
-
- {XFS_INHERIT_NODFRG, "inherit_nodefrag", &xfs_params.inherit_nodfrg.val,
- sizeof(int), 0644, NULL, &proc_dointvec_minmax,
- &sysctl_intvec, NULL,
- &xfs_params.inherit_nodfrg.min, &xfs_params.inherit_nodfrg.max},
+ {
+  .ctl_name = XFS_RESTRICT_CHOWN,
+  .procname = "restrict_chown",
+  .data  = &xfs_params.restrict_chown.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.restrict_chown.min,
+  .extra2  = &xfs_params.restrict_chown.max
+ },
+ {
+  .ctl_name = XFS_SGID_INHERIT,
+  .procname = "irix_sgid_inherit",
+  .data  = &xfs_params.sgid_inherit.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.sgid_inherit.min,
+  .extra2  = &xfs_params.sgid_inherit.max
+ },
+ {
+  .ctl_name = XFS_SYMLINK_MODE,
+  .procname = "irix_symlink_mode",
+  .data  = &xfs_params.symlink_mode.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.symlink_mode.min,
+  .extra2  = &xfs_params.symlink_mode.max
+ },
+ {
```

```
+  .ctl_name = XFS_PANIC_MASK,
+  .procname = "panic_mask",
+  .data  = &xfs_params.panic_mask.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler =  &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.panic_mask.min,
+  .extra2  = &xfs_params.panic_mask.max
+ },

+ {
+  .ctl_name = XFS_ERRLEVEL,
+  .procname = "error_level",
+  .data  = &xfs_params.error_level.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.error_level.min,
+  .extra2  = &xfs_params.error_level.max
+ },
+ {
+  .ctl_name = XFS_SYNCD_TIMER,
+  .procname = "xfssyncd_centisecs",
+  .data  = &xfs_params.syncd_timer.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.syncd_timer.min,
+  .extra2  = &xfs_params.syncd_timer.max
+ },
+ {
+  .ctl_name = XFS_INHERIT_SYNC,
+  .procname = "inherit_sync",
+  .data  = &xfs_params.inherit_sync.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.inherit_sync.min,
+  .extra2  = &xfs_params.inherit_sync.max
+ },
+ {
+  .ctl_name = XFS_INHERIT_NODUMP,
+  .procname = "inherit_nodump",
+  .data  = &xfs_params.inherit_nodump.val,
```

```
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec, NULL,
+  .extra1  = &xfs_params.inherit_nodump.min,
+  .extra2  = &xfs_params.inherit_nodump.max
+ },
+ {
+  .ctl_name = XFS_INHERIT_NOATIME,
+  .procname = "inherit_noatime",
+  .data  = &xfs_params.inherit_noatim.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec, NULL,
+  .extra1  = &xfs_params.inherit_noatim.min,
+  .extra2  = &xfs_params.inherit_noatim.max
+ },
+ {
+  .ctl_name = XFS_BUF_TIMER,
+  .procname = "xfsbufd_centisecs",
+  .data  = &xfs_params.xfs_buf_timer.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.xfs_buf_timer.min,
+  .extra2  = &xfs_params.xfs_buf_timer.max
+ },
+ {
+  .ctl_name = XFS_BUF_AGE,
+  .procname = "age_buffer_centisecs",
+  .data  = &xfs_params.xfs_buf_age.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec, NULL,
+  .extra1  = &xfs_params.xfs_buf_age.min,
+  .extra2  = &xfs_params.xfs_buf_age.max
+ },
+ {
+  .ctl_name = XFS_INHERIT_NOSYM,
+  .procname = "inherit_nosymlinks",
+  .data  = &xfs_params.inherit_nosym.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
```

```
+  .extra1  = &xfs_params.inherit_nosym.min,
+  .extra2  = &xfs_params.inherit_nosym.max
+ },
+ {
+  .ctl_name = XFS_ROTORSTEP,
+  .procname = "rotorstep",
+  .data  = &xfs_params.rotorstep.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.rotorstep.min,
+  .extra2  = &xfs_params.rotorstep.max
+ },
+ {
+  .ctl_name = XFS_INHERIT_NODFRG,
+  .procname = "inherit_nodefrag",
+  .data  = &xfs_params.inherit_nodfrg.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.inherit_nodfrg.min,
+  .extra2  = &xfs_params.inherit_nodfrg.max
+ },
   /* please keep this the last entry */
  #ifdef CONFIG_PROC_FS
- {XFS_STATS_CLEAR, "stats_clear", &xfs_params.stats_clear.val,
- sizeof(int), 0644, NULL, &xfs_stats_clear_proc_handler,
- &sysctl_intvec, NULL,
- &xfs_params.stats_clear.min, &xfs_params.stats_clear.max},
+ {
+  .ctl_name = XFS_STATS_CLEAR,
+  .procname = "stats_clear",
+  .data  = &xfs_params.stats_clear.val,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler =  &xfs_stats_clear_proc_handler,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xfs_params.stats_clear.min,
+  .extra2  = &xfs_params.stats_clear.max
+ },
  #endif /* CONFIG_PROC_FS */

- {0}
+ {}
  };
```

```
 STATIC ctl_table xfs_dir_table[] = {
- {FS_XFS, "xfs", NULL, 0, 0555, xfs_table},
- {0}
+ {
+  .ctl_name = FS_XFS,
+  .procname = "xfs",
+  .mode  = 0555,
+  .child  = xfs_table
+ },
+ {}
 };

 STATIC ctl_table xfs_root_table[] = {
- {CTL_FS, "fs",  NULL, 0, 0555, xfs_dir_table},
- {0}
+ {
+  .ctl_name = CTL_FS,
+  .procname = "fs",
+  .mode  = 0555,
+  .child  = xfs_dir_table
+ },
+ {}
 };

 void
--
1.4.4.1.g278f
```

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

Subject: [PATCH 15/59] sysctl: scsi remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:20 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/scsi/scsi_sysctl.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/drivers/scsi/scsi_sysctl.c b/drivers/scsi/scsi_sysctl.c
index 04d06c2..b16b775 100644
--- a/drivers/scsi/scsi_sysctl.c

```
+++ b/drivers/scsi/scsi_sysctl.c
@@ -41,7 +41,7 @@ static struct ctl_table_header *scsi_table_header;

 int __init scsi_init_sysctl(void)
 {
- scsi_table_header = register_sysctl_table(scsi_root_table, 1);
+ scsi_table_header = register_sysctl_table(scsi_root_table, 0);
  if (!scsi_table_header)
   return -ENOMEM;
  return 0;
--
1.4.4.1.g278f
```

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

Subject: [PATCH 16/59] sysctl: md Remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:21 GMT
View Forum Message <> Reply to Message

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

The sysctls used by the md driver are have unique binary numbers
so remove the insert_at_head flag as it serves no useful purpose.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/md/md.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

```
diff --git a/drivers/md/md.c b/drivers/md/md.c
index d1cb45f..966e8be 100644
--- a/drivers/md/md.c
+++ b/drivers/md/md.c
@@ -5551,7 +5551,7 @@ static int __init md_init(void)
        md_probe, NULL, NULL);

  register_reboot_notifier(&md_notifier);
- raid_table_header = register_sysctl_table(raid_root_table, 1);
+ raid_table_header = register_sysctl_table(raid_root_table, 0);

  md_geninit();
  return (0);
--
1.4.4.1.g278f
```

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

Subject: [PATCH 17/59] sysctl: mac_hid remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:22 GMT
View Forum Message <> Reply to Message

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

With unique sysctl binary numbers setting insert_at_head is pointless.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/macintosh/mac_hid.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/drivers/macintosh/mac_hid.c b/drivers/macintosh/mac_hid.c
index ee6b4ca..c676740 100644
--- a/drivers/macintosh/mac_hid.c
+++ b/drivers/macintosh/mac_hid.c
@@ -138,7 +138,7 @@ int __init mac_hid_init(void)
   return err;

 #if defined(CONFIG_SYSCTL)
- mac_hid_sysctl_header = register_sysctl_table(mac_hid_root_dir, 1);
+ mac_hid_sysctl_header = register_sysctl_table(mac_hid_root_dir, 0);
 #endif /* CONFIG_SYSCTL */

  return 0;
--
1.4.4.1.g278f

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

Subject: [PATCH 18/59] sysctl: ipmi remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:23 GMT
View Forum Message <> Reply to Message

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

With unique sysctl binary numbers setting insert_at_head is pointless.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/char/ipmi/ipmi_poweroff.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/drivers/char/ipmi/ipmi_poweroff.c b/drivers/char/ipmi/ipmi_poweroff.c
index 9d23136..b3ae65e 100644
--- a/drivers/char/ipmi/ipmi_poweroff.c
+++ b/drivers/char/ipmi/ipmi_poweroff.c
@@ -686,7 +686,7 @@ static int ipmi_poweroff_init (void)
   printk(KERN_INFO PFX "Power cycle is enabled.\n");

 #ifdef CONFIG_PROC_FS
- ipmi_table_header = register_sysctl_table(ipmi_root_table, 1);
+ ipmi_table_header = register_sysctl_table(ipmi_root_table, 0);
  if (!ipmi_table_header) {
   printk(KERN_ERR PFX "Unable to register powercycle sysctl\n");
   rv = -ENOMEM;
--
1.4.4.1.g278f

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

Subject: [PATCH 19/59] sysctl: cdrom remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:24 GMT
View Forum Message <> Reply to Message

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

With unique binary sysctl numbers setting insert_at_head to
override other sysctl entries is pointless.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/cdrom/cdrom.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/drivers/cdrom/cdrom.c b/drivers/cdrom/cdrom.c
index 3105ddd..f0a6801 100644
--- a/drivers/cdrom/cdrom.c
+++ b/drivers/cdrom/cdrom.c

```
@@ -3553,7 +3553,7 @@ static void cdrom_sysctl_register(void)
  if (initialized == 1)
   return;

- cdrom_sysctl_header = register_sysctl_table(cdrom_root_table, 1);
+ cdrom_sysctl_header = register_sysctl_table(cdrom_root_table, 0);
  if (cdrom_root_table->ctl_name && cdrom_root_table->child->de)
   cdrom_root_table->child->de->owner = THIS_MODULE;

--
1.4.4.1.g278f
```

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

Subject: [PATCH 20/59] sysctl: cdrom Don't set de->owner
Posted by ebiederm on Tue, 16 Jan 2007 16:39:25 GMT
View Forum Message <> Reply to Message

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

There is no need for open files in /proc/sys/XXX to hold
a reference count on the module that provides the file
to prevent module unload races. While there is code active
in the module p->used in the sysctl_table_header is incremented,
preventing the sysctl from being unregisted. Once the
sysctl is unregistered it cannot be found. Open files
are also not a problem as they revalidate the sysctl information
and bump p->used before accessing module code.

So setting de->owner is unnecessary, makes for a bad example
and gets in my way of removing ctl_table->de.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/cdrom/cdrom.c | 2 --
 1 files changed, 0 insertions(+), 2 deletions(-)

```
diff --git a/drivers/cdrom/cdrom.c b/drivers/cdrom/cdrom.c
index f0a6801..14f72c4 100644
--- a/drivers/cdrom/cdrom.c
+++ b/drivers/cdrom/cdrom.c
@@ -3554,8 +3554,6 @@ static void cdrom_sysctl_register(void)
   return;
```

```
  cdrom_sysctl_header = register_sysctl_table(cdrom_root_table, 0);
- if (cdrom_root_table->ctl_name && cdrom_root_table->child->de)
-  cdrom_root_table->child->de->owner = THIS_MODULE;

  /* set the defaults */
  cdrom_sysctl_settings.autoclose = autoclose;
--
1.4.4.1.g278f
```

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

Subject: [PATCH 21/59] sysctl: Move CTL_PM into sysctl.h where it belongs.
Posted by ebiederm on Tue, 16 Jan 2007 16:39:26 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 arch/frv/kernel/pm.c   |    1 -
 include/linux/sysctl.h |    1 +
 2 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/arch/frv/kernel/pm.c b/arch/frv/kernel/pm.c
index ee677ce..6b76466 100644
--- a/arch/frv/kernel/pm.c
+++ b/arch/frv/kernel/pm.c
@@ -125,7 +125,6 @@ unsigned long sleep_phys_sp(void *sp)
  * Use a temporary sysctl number. Horrid, but will be cleaned up in 2.6
  * when all the PM interfaces exist nicely.
  */
-#define CTL_PM 9899
 #define CTL_PM_SUSPEND 1
 #define CTL_PM_CMODE 2
 #define CTL_PM_P0 4
diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 54a9cf5..e7c40b6 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -71,6 +71,7 @@ enum
  CTL_ABI=9,  /* Binary emulation */
  CTL_CPU=10,  /* CPU stuff (speed scaling, etc) */
  CTL_SUNRPC=7249, /* sunrpc debug */
+ CTL_PM=9899,  /* frv power management */
```

```
 };

 /* CTL_BUS names: */
--
1.4.4.1.g278f
```

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

Subject: [PATCH 22/59] sysctl: frv pm remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:27 GMT
View Forum Message <> Reply to Message

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

With unique binary numbers setting insert_at_head to
insert yourself at the head of sysctl list and thus override
existing sysctl entries serves no point.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/frv/kernel/pm.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

```
diff --git a/arch/frv/kernel/pm.c b/arch/frv/kernel/pm.c
index 6b76466..c1840d6 100644
--- a/arch/frv/kernel/pm.c
+++ b/arch/frv/kernel/pm.c
@@ -419,7 +419,7 @@ static struct ctl_table pm_dir_table[] =
  */
 static int __init pm_init(void)
 {
- register_sysctl_table(pm_dir_table, 1);
+ register_sysctl_table(pm_dir_table, 0);
  return 0;
 }

--
1.4.4.1.g278f
```

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

Subject: [PATCH 23/59] sysctl: Move CTL_FRV into sysctl.h where it belongs
Posted by ebiederm on Tue, 16 Jan 2007 16:39:28 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/frv/kernel/sysctl.c | 1 -
 include/linux/sysctl.h | 1 +
 2 files changed, 1 insertions(+), 1 deletions(-)

```
diff --git a/arch/frv/kernel/sysctl.c b/arch/frv/kernel/sysctl.c
index ce67680..2f4da32 100644
--- a/arch/frv/kernel/sysctl.c
+++ b/arch/frv/kernel/sysctl.c
@@ -186,7 +186,6 @@ static struct ctl_table frv_table[] =
  * Use a temporary sysctl number. Horrid, but will be cleaned up in 2.6
  * when all the PM interfaces exist nicely.
  */
-#define CTL_FRV 9898
 static struct ctl_table frv_dir_table[] =
 {
  {CTL_FRV, "frv", NULL, 0, 0555, frv_table},
diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index e7c40b6..71c16b4 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -72,6 +72,7 @@ enum
  CTL_CPU=10,  /* CPU stuff (speed scaling, etc) */
  CTL_SUNRPC=7249, /* sunrpc debug */
  CTL_PM=9899,  /* frv power management */
+ CTL_FRV=9898,  /* frv specific sysctls */
 };

 /* CTL_BUS names: */
--
1.4.4.1.g278f
```

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

Subject: [PATCH 24/59] sysctl: frv remove unnecessary insert_at_head flag
Posted by ebiederm on Tue, 16 Jan 2007 16:39:29 GMT
View Forum Message <> Reply to Message

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

Since the binary sysctl numbers are unique putting the registered
sysctls at the head of the sysctl list where they can override
existing sysctls serves no useful purpose.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/frv/kernel/sysctl.c | 2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/arch/frv/kernel/sysctl.c b/arch/frv/kernel/sysctl.c
index 2f4da32..37528eb 100644
--- a/arch/frv/kernel/sysctl.c
+++ b/arch/frv/kernel/sysctl.c
@@ -197,7 +197,7 @@ static struct ctl_table frv_dir_table[] =
  */
 static int __init frv_sysctl_init(void)
 {
- register_sysctl_table(frv_dir_table, 1);
+ register_sysctl_table(frv_dir_table, 0);
  return 0;
 }

--
1.4.4.1.g278f

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

Subject: [PATCH 25/59] sysctl: C99 convert arch/frv/kernel/pm.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:30 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/frv/kernel/pm.c | 50 ++-------
 1 files changed, 43 insertions(+), 7 deletions(-)

diff --git a/arch/frv/kernel/pm.c b/arch/frv/kernel/pm.c
index c1840d6..aa50333 100644
--- a/arch/frv/kernel/pm.c
+++ b/arch/frv/kernel/pm.c

```
@@ -401,17 +401,53 @@ static int cm_sysctl(ctl_table *table, int __user *name, int nlen,

 static struct ctl_table pm_table[] =
 {
- {CTL_PM_SUSPEND, "suspend", NULL, 0, 0200, NULL, &sysctl_pm_do_suspend},
- {CTL_PM_CMODE, "cmode", &clock_cmode_current, sizeof(int), 0644, NULL, &cmode_procctl,
&cmode_sysctl, NULL},
- {CTL_PM_P0, "p0", &clock_p0_current, sizeof(int), 0644, NULL, &p0_procctl, &p0_sysctl,
NULL},
- {CTL_PM_CM, "cm", &clock_cm_current, sizeof(int), 0644, NULL, &cm_procctl, &cm_sysctl,
NULL},
- {0}
+ {
+  .ctl_name = CTL_PM_SUSPEND,
+  .procname = "suspend",
+  .data  = NULL,
+  .maxlen  = 0,
+  .mode  = 0200,
+  .proc_handler = &sysctl_pm_do_suspend,
+ },
+ {
+  .ctl_name = CTL_PM_CMODE,
+  .procname = "cmode",
+  .data  = &clock_cmode_current,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &cmode_procctl,
+  .strategy = &cmode_sysctl,
+ },
+ {
+  .ctl_name = CTL_PM_P0,
+  .procname = "p0",
+  .data  = &clock_p0_current,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &p0_procctl,
+  .strategy = &p0_sysctl,
+ },
+ {
+  .ctl_name = CTL_PM_CM,
+  .procname = "cm",
+  .data  = &clock_cm_current,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &cm_procctl,
+  .strategy = &cm_sysctl,
+ },
+ { .ctl_name = 0}
```

```
 };

 static struct ctl_table pm_dir_table[] =
 {
- {CTL_PM, "pm", NULL, 0, 0555, pm_table},
- {0}
+ {
+  .ctl_name = CTL_PM,
+  .procname = "pm",
+  .mode  = 0555,
+  .child  = pm_table,
+ },
+ { .ctl_name = 0}
 };

 /*
--
1.4.4.1.g278f
```

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

Subject: [PATCH 26/59] sysctl: C99 convert arch/frv/kernel/sysctl.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:31 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/frv/kernel/sysctl.c | 29 +++++++++++++++++++++++-----
 1 files changed, 24 insertions(+), 5 deletions(-)

```
diff --git a/arch/frv/kernel/sysctl.c b/arch/frv/kernel/sysctl.c
index 37528eb..577ad16 100644
--- a/arch/frv/kernel/sysctl.c
+++ b/arch/frv/kernel/sysctl.c
@@ -175,11 +175,25 @@ static int procctl_frv_pin_cxnr(ctl_table *table, int write, struct file *filp,
  */
 static struct ctl_table frv_table[] =
 {
- { 1, "cache-mode", NULL, 0, 0644, NULL, &procctl_frv_cachemode },
+ {
+  .ctl_name  = 1,
+  .procname  = "cache-mode",
```

```
+  .data  = NULL,
+  .maxlen  = 0,
+  .mode  = 0644,
+  .proc_handler = &procctl_frv_cachemode,
+ },
 #ifdef CONFIG_MMU
- { 2, "pin-cxnr", NULL, 0, 0644, NULL, &procctl_frv_pin_cxnr },
+ {
+  .ctl_name = 2,
+  .procname = "pin-cxnr",
+  .data  = NULL,
+  .maxlen  = 0,
+  .mode  = 0644,
+  .proc_handler = &procctl_frv_pin_cxnr
+ },
 #endif
- { 0 }
+ {}
 };

 /*
@@ -188,8 +202,13 @@ static struct ctl_table frv_table[] =
  */
 static struct ctl_table frv_dir_table[] =
 {
- {CTL_FRV, "frv", NULL, 0, 0555, frv_table},
- {0}
+ {
+  .ctl_name = CTL_FRV,
+  .procname = "frv",
+  .mode   = 0555,
+  .child  = frv_table
+ },
+ {}
 };

 /*
--
1.4.4.1.g278f
```

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

Subject: [PATCH 27/59] sysctl: sn Remove sysctl ABI BREAKAGE

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

By not using the enumeration in sysctl.h (or even understanding it)
the SN platform placed their arch specific xpc directory on top of
CTL_KERN and only because they didn't have 4 entries in their xpc
directory got lucky and didn't break glibc.

This is totally irresponsible. So this patch entirely removes
sys_sysctl support from their sysctl code. Hopefully they
don't have ascii name conflicts as well.

And now that they have no ABI numbers add them to the end
instead of the sysctl list instead of the head so nothing
else will be overridden.

Cc: Tony Luck <tony.luck@intel.com>
Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/ia64/sn/kernel/xpc_main.c | 12 ++++++------
 1 files changed, 6 insertions(+), 6 deletions(-)

diff --git a/arch/ia64/sn/kernel/xpc_main.c b/arch/ia64/sn/kernel/xpc_main.c
index 7a387d2..24adb75 100644
--- a/arch/ia64/sn/kernel/xpc_main.c
+++ b/arch/ia64/sn/kernel/xpc_main.c
@@ -101,7 +101,7 @@ static int xpc_disengage_request_max_timelimit = 120;

 static ctl_table xpc_sys_xpc_hb_dir[] = {
  {
- 1,
+ CTL_UNNUMBERED,
   "hb_interval",
   &xpc_hb_interval,
   sizeof(int),
@@ -114,7 +114,7 @@ static ctl_table xpc_sys_xpc_hb_dir[] = {
   &xpc_hb_max_interval
  },
  {
- 2,
+ CTL_UNNUMBERED,
   "hb_check_interval",
   &xpc_hb_check_interval,
   sizeof(int),
@@ -130,7 +130,7 @@ static ctl_table xpc_sys_xpc_hb_dir[] = {
 };
 static ctl_table xpc_sys_xpc_dir[] = {

```
  {
-  1,
+  CTL_UNNUMBERED,
   "hb",
   NULL,
   0,
@@ -138,7 +138,7 @@ static ctl_table xpc_sys_xpc_dir[] = {
   xpc_sys_xpc_hb_dir
  },
  {
-  2,
+  CTL_UNNUMBERED,
   "disengage_request_timelimit",
   &xpc_disengage_request_timelimit,
   sizeof(int),
@@ -154,7 +154,7 @@ static ctl_table xpc_sys_xpc_dir[] = {
 };
 static ctl_table xpc_sys_dir[] = {
  {
-  1,
+  CTL_UNNUMBERED,
   "xpc",
   NULL,
   0,
@@ -1251,7 +1251,7 @@ xpc_init(void)
  snprintf(xpc_part->bus_id, BUS_ID_SIZE, "part");
  snprintf(xpc_chan->bus_id, BUS_ID_SIZE, "chan");

- xpc_sysctl = register_sysctl_table(xpc_sys_dir, 1);
+ xpc_sysctl = register_sysctl_table(xpc_sys_dir, 0);

  /*
   * The first few fields of each entry of xpc_partitions[] need to
--
1.4.4.1.g278f
```

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

Subject: [PATCH 28/59] sysctl: C99 Convert arch/ia64/sn/kernel/xpc_main.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:33 GMT

View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/ia64/sn/kernel/xpc_main.c | 86 ++++++++++++++++++++++----------------
 1 files changed, 38 insertions(+), 48 deletions(-)

diff --git a/arch/ia64/sn/kernel/xpc_main.c b/arch/ia64/sn/kernel/xpc_main.c
index 24adb75..e04f7b5 100644
--- a/arch/ia64/sn/kernel/xpc_main.c
+++ b/arch/ia64/sn/kernel/xpc_main.c
@@ -101,67 +101,57 @@ static int xpc_disengage_request_max_timelimit = 120;

 static ctl_table xpc_sys_xpc_hb_dir[] = {
  {
- CTL_UNNUMBERED,
- "hb_interval",
- &xpc_hb_interval,
- sizeof(int),
- 0644,
- NULL,
- &proc_dointvec_minmax,
- &sysctl_intvec,
- NULL,
- &xpc_hb_min_interval,
- &xpc_hb_max_interval
+ .ctl_name = CTL_UNNUMBERED,
+ .procname = "hb_interval",
+ .data = &xpc_hb_interval,
+ .maxlen = sizeof(int),
+ .mode = 0644,
+ .proc_handler = &proc_dointvec_minmax,
+ .strategy = &sysctl_intvec,
+ .extra1 = &xpc_hb_min_interval,
+ .extra2 = &xpc_hb_max_interval
  },
  {
- CTL_UNNUMBERED,
- "hb_check_interval",
- &xpc_hb_check_interval,
- sizeof(int),
- 0644,
- NULL,
- &proc_dointvec_minmax,
- &sysctl_intvec,
- NULL,
- &xpc_hb_check_min_interval,
- &xpc_hb_check_max_interval
+ .ctl_name = CTL_UNNUMBERED,
+ .procname = "hb_check_interval",

```
+  .data  = &xpc_hb_check_interval,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xpc_hb_check_min_interval,
+  .extra2  = &xpc_hb_check_max_interval
   },
- {0}
+ {}
 };
 static ctl_table xpc_sys_xpc_dir[] = {
   {
-  CTL_UNNUMBERED,
-  "hb",
-  NULL,
-  0,
-  0555,
-  xpc_sys_xpc_hb_dir
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "hb",
+  .mode  = 0555,
+  .child  = xpc_sys_xpc_hb_dir
   },
   {
-  CTL_UNNUMBERED,
-  "disengage_request_timelimit",
-  &xpc_disengage_request_timelimit,
-  sizeof(int),
-  0644,
-  NULL,
-  &proc_dointvec_minmax,
-  &sysctl_intvec,
-  NULL,
-  &xpc_disengage_request_min_timelimit,
-  &xpc_disengage_request_max_timelimit
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "disengage_request_timelimit",
+  .data  = &xpc_disengage_request_timelimit,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .strategy = &sysctl_intvec,
+  .extra1  = &xpc_disengage_request_min_timelimit,
+  .extra2  = &xpc_disengage_request_max_timelimit
   },
- {0}
+ {}
```

```
 };
 static ctl_table xpc_sys_dir[] = {
  {
-  CTL_UNNUMBERED,
-  "xpc",
-  NULL,
-  0,
-  0555,
-  xpc_sys_xpc_dir
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "xpc",
+  .mode  = 0555,
+  .child  = xpc_sys_xpc_dir
  },
- {0}
+ {}
 };
 static struct ctl_table_header *xpc_sysctl;

--
1.4.4.1.g278f
```

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

Subject: [PATCH 29/59] sysctl: C99 convert arch/ia64/kernel/perfmon and remove ABI breakage

Posted by ebiederm on Tue, 16 Jan 2007 16:39:34 GMT
View Forum Message <> Reply to Message

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

This convers the sysctl ctl_tables to use C99 initializers.
While I was looking at it I discovered it was using a portion of
the sysctl binary addresses space under CTL_KERN KERN_OSTYPE
which was completely inappropriate. So I completely removed
all of the sysctl binary names, to remove and avoid the ABI conflict.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/ia64/kernel/perfmon.c | 56 ++++++++++++++++++++++++++++++++++++-------
 1 files changed, 47 insertions(+), 9 deletions(-)

diff --git a/arch/ia64/kernel/perfmon.c b/arch/ia64/kernel/perfmon.c
index aa94f60..8c679ab 100644

```
--- a/arch/ia64/kernel/perfmon.c
+++ b/arch/ia64/kernel/perfmon.c
@@ -521,19 +521,57 @@ pfm_sysctl_t pfm_sysctl;
 EXPORT_SYMBOL(pfm_sysctl);

 static ctl_table pfm_ctl_table[]={
- {1, "debug", &pfm_sysctl.debug, sizeof(int), 0666, NULL, &proc_dointvec, NULL,},
- {2, "debug_ovfl", &pfm_sysctl.debug_ovfl, sizeof(int), 0666, NULL, &proc_dointvec, NULL,},
- {3, "fastctxsw", &pfm_sysctl.fastctxsw, sizeof(int), 0600, NULL, &proc_dointvec, NULL,},
- {4, "expert_mode", &pfm_sysctl.expert_mode, sizeof(int), 0600, NULL, &proc_dointvec, NULL,},
- { 0, },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "debug",
+  .data  = &pfm_sysctl.debug,
+  .maxlen  = sizeof(int),
+  .mode  = 0666,
+  .proc_handler = &proc_dointvec,
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "debug_ovfl",
+  .data  = &pfm_sysctl.debug_ovfl,
+  .maxlen  = sizeof(int),
+  .mode  = 0666,
+  .proc_handler = &proc_dointvec,
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "fastctxsw",
+  .data  = &pfm_sysctl.fastctxsw,
+  .maxlen  = sizeof(int),
+  .mode  = 0600,
+  .proc_handler =  &proc_dointvec,
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "expert_mode",
+  .data  = &pfm_sysctl.expert_mode,
+  .maxlen  = sizeof(int),
+  .mode  = 0600,
+  .proc_handler = &proc_dointvec,
+ },
+ {}
 };
 static ctl_table pfm_sysctl_dir[] = {
- {1, "perfmon", NULL, 0, 0755, pfm_ctl_table, },
-  {0,},
```

```
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "perfmon",
+  .mode  = 0755,
+  .child  = pfm_ctl_table,
+ },
+  {}
 };
 static ctl_table pfm_sysctl_root[] = {
- {1, "kernel", NULL, 0, 0755, pfm_sysctl_dir, },
-  {0,},
+ {
+  .ctl_name = CTL_KERN,
+  .procname = "kernel",
+  .mode  = 0755,
+  .child  = pfm_sysctl_dir,
+ },
+  {}
 };
 static struct ctl_table_header *pfm_sysctl_header;

--
1.4.4.1.g278f
```

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

Subject: [PATCH 30/59] sysctl: mips/au1000 Remove sys_sysctl support
Posted by ebiederm on Tue, 16 Jan 2007 16:39:35 GMT
View Forum Message <> Reply to Message

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

The assignment of binary numbers for sys_sysctl use was in
shambles and despite requiring methods. Nothing was implemented
on the sys_sysctl side.

So this patch gives a mercy killing to the sys_sysctl support for
powermanagment on mips/au1000.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/mips/au1000/common/power.c | 16 +++++-----------
 1 files changed, 5 insertions(+), 11 deletions(-)

```
diff --git a/arch/mips/au1000/common/power.c b/arch/mips/au1000/common/power.c
index 7504a63..b531ab7 100644
--- a/arch/mips/au1000/common/power.c
+++ b/arch/mips/au1000/common/power.c
@@ -62,12 +62,6 @@ extern unsigned long save_local_and_disable(int controller);
 extern void restore_local_and_enable(int controller, unsigned long mask);
 extern void local_enable_irq(unsigned int irq_nr);

-/* Quick acpi hack. This will have to change! */
-#define CTL_ACPI 9999
-#define ACPI_S1_SLP_TYP 19
-#define ACPI_SLEEP 21
-
-
 static DEFINE_SPINLOCK(pm_lock);

 /* We need to save/restore a bunch of core registers that are
@@ -425,14 +419,14 @@ static int pm_do_freq(ctl_table * ctl, int write, struct file *file,


 static struct ctl_table pm_table[] = {
- {ACPI_S1_SLP_TYP, "suspend", NULL, 0, 0600, NULL, &pm_do_suspend},
- {ACPI_SLEEP, "sleep", NULL, 0, 0600, NULL, &pm_do_sleep},
- {CTL_ACPI, "freq", NULL, 0, 0600, NULL, &pm_do_freq},
+ {CTL_UNNUMBERED, "suspend", NULL, 0, 0600, NULL, &pm_do_suspend},
+ {CTL_UNNUMBERED, "sleep", NULL, 0, 0600, NULL, &pm_do_sleep},
+ {CTL_UNNUMBERED, "freq", NULL, 0, 0600, NULL, &pm_do_freq},
  {0}
 };

 static struct ctl_table pm_dir_table[] = {
- {CTL_ACPI, "pm", NULL, 0, 0555, pm_table},
+ {CTL_UNNUMBERED, "pm", NULL, 0, 0555, pm_table},
  {0}
 };

@@ -441,7 +435,7 @@ static struct ctl_table pm_dir_table[] = {
  */
 static int __init pm_init(void)
 {
- register_sysctl_table(pm_dir_table, 1);
+ register_sysctl_table(pm_dir_table, 0);
  return 0;
 }

--
1.4.4.1.g278f
```

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

Subject: [PATCH 31/59] sysctl: C99 convert the ctl_tables in arch/mips/au1000/common/power.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:36 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/mips/au1000/common/power.c | 38 ++++++++++++++++++++++++++++++++------
 1 files changed, 32 insertions(+), 6 deletions(-)

diff --git a/arch/mips/au1000/common/power.c b/arch/mips/au1000/common/power.c
index b531ab7..31256b8 100644
--- a/arch/mips/au1000/common/power.c
+++ b/arch/mips/au1000/common/power.c
@@ -419,15 +419,41 @@ static int pm_do_freq(ctl_table * ctl, int write, struct file *file,

 static struct ctl_table pm_table[] = {
-{CTL_UNNUMBERED, "suspend", NULL, 0, 0600, NULL, &pm_do_suspend},
-{CTL_UNNUMBERED, "sleep", NULL, 0, 0600, NULL, &pm_do_sleep},
-{CTL_UNNUMBERED, "freq", NULL, 0, 0600, NULL, &pm_do_freq},
-{0}
+{
+ .ctl_name = CTL_UNNUMBERED,
+ .procname = "suspend",
+ .data = NULL,
+ .maxlen = 0,
+ .mode = 0600,
+ .proc_handler = &pm_do_suspend
+},
+{
+ .ctl_name = CTL_UNNUMBERED,
+ .procname = "sleep",
+ .data = NULL,
+ .maxlen = 0,
+ .mode = 0600,
+ .proc_handler = &pm_do_sleep
+},
+{
+ .ctl_name = CTL_UNNUMBERED,

```
+  .procname = "freq",
+  .data  = NULL,
+  .maxlen  = 0,
+  .mode  = 0600,
+  .proc_handler = &pm_do_freq
+ },
+ {}
 };

 static struct ctl_table pm_dir_table[] = {
- {CTL_UNNUMBERED, "pm", NULL, 0, 0555, pm_table},
- {0}
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "pm",
+  .mode  = 0555,
+  .child  = pm_table
+ },
+ {}
 };

 /*
--
1.4.4.1.g278f
```

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

Subject: [PATCH 32/59] sysctl: C99 convert arch/mips/lasat/sysctl.c and remove ABI breakage.

Posted by ebiederm on Tue, 16 Jan 2007 16:39:37 GMT
View Forum Message <> Reply to Message

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

While C99 converting the ctl_table initializers I realized
that the binary sysctl numbers were in conflict with the binary
values under CTL_KERN. Including CTL_KERN KERN_VERSION as used
by glibc. So I just removed the sysctl binary interface for these
values, as it was unsupportable.

Luckily these sysctl were inserted at the end of the
sysctl list so this bug was not visible to userspace.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 arch/mips/lasat/sysctl.c |  145 ++++++++++++++++++++++++++++++++---------
 1 files changed, 116 insertions(+), 29 deletions(-)

diff --git a/arch/mips/lasat/sysctl.c b/arch/mips/lasat/sysctl.c
index 1287835..c04e82f 100644
--- a/arch/mips/lasat/sysctl.c
+++ b/arch/mips/lasat/sysctl.c
@@ -302,42 +302,129 @@ extern int lasat_boot_to_service;
 #ifdef CONFIG_SYSCTL

 static ctl_table lasat_table[] = {
- {LASAT_CPU_HZ, "cpu-hz", &lasat_board_info.li_cpu_hz, sizeof(int),
-  0444, NULL, &proc_dointvec, &sysctl_intvec},
- {LASAT_BUS_HZ, "bus-hz", &lasat_board_info.li_bus_hz, sizeof(int),
-  0444, NULL, &proc_dointvec, &sysctl_intvec},
- {LASAT_MODEL, "bmid", &lasat_board_info.li_bmid, sizeof(int),
-  0444, NULL, &proc_dointvec, &sysctl_intvec},
- {LASAT_PRID, "prid", &lasat_board_info.li_prid, sizeof(int),
-  0644, NULL, &proc_lasat_eeprom_value, &sysctl_lasat_eeprom_value},
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "cpu-hz",
+  .data  = &lasat_board_info.li_cpu_hz,
+  .maxlen  = sizeof(int),
+  .mode  = 0444,
+  .proc_handler = &proc_dointvec,
+  .strategy = &sysctl_intvec
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "bus-hz",
+  .data  = &lasat_board_info.li_bus_hz,
+  .maxlen  = sizeof(int),
+  .mode  = 0444,
+  .proc_handler = &proc_dointvec,
+  .strategy = &sysctl_intvec
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "bmid",
+  .data  = &lasat_board_info.li_bmid,
+  .maxlen  = sizeof(int),
+  .mode  = 0444,
+  .proc_handler = &proc_dointvec,
+  .strategy = &sysctl_intvec
+ },
+ {
```

```
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "prid",
+  .data  = &lasat_board_info.li_prid,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_lasat_eeprom_value,
+  .strategy = &sysctl_lasat_eeprom_value
+ },
 #ifdef CONFIG_INET
- {LASAT_IPADDR, "ipaddr", &lasat_board_info.li_eeprom_info.ipaddr, sizeof(int),
-  0644, NULL, &proc_lasat_ip, &sysctl_lasat_intvec},
- {LASAT_NETMASK, "netmask", &lasat_board_info.li_eeprom_info.netmask, sizeof(int),
-  0644, NULL, &proc_lasat_ip, &sysctl_lasat_intvec},
- {LASAT_BCAST, "bcastaddr", &lasat_bcastaddr,
-  sizeof(lasat_bcastaddr), 0600, NULL,
-  &proc_dostring, &sysctl_string},
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "ipaddr",
+  .data  = &lasat_board_info.li_eeprom_info.ipaddr,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_lasat_ip,
+  .strategy = &sysctl_lasat_intvec
+ },
+ {
+  .ctl_name = LASAT_NETMASK,
+  .procname = "netmask",
+  .data  = &lasat_board_info.li_eeprom_info.netmask,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_lasat_ip,
+  .strategy = &sysctl_lasat_intvec
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "bcastaddr",
+  .data  = &lasat_bcastaddr,
+  .maxlen  = sizeof(lasat_bcastaddr),
+  .mode  = 0600,
+  .proc_handler = &proc_dostring,
+  .strategy = &sysctl_string
+ },
 #endif
- {LASAT_PASSWORD, "passwd_hash", &lasat_board_info.li_eeprom_info.passwd_hash,
sizeof(lasat_board_info.li_eeprom_info.passwd_hash),
-  0600, NULL, &proc_dolasatstring, &sysctl_lasatstring},
- {LASAT_SBOOT, "boot-service", &lasat_boot_to_service, sizeof(int),
```

```
-  0644, NULL, &proc_dointvec, &sysctl_intvec},
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "passwd_hash",
+  .data  = &lasat_board_info.li_eeprom_info.passwd_hash,
+  .maxlen  = sizeof(lasat_board_info.li_eeprom_info.passwd_hash),
+  .mode  = 0600,
+  .proc_handler = &proc_dolasatstring,
+  .strategy = &sysctl_lasatstring
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "boot-service",
+  .data  = &lasat_boot_to_service,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec,
+  .strategy = &sysctl_intvec
+ },
 #ifdef CONFIG_DS1603
- {LASAT_RTC, "rtc", &rtctmp, sizeof(int),
-  0644, NULL, &proc_dolasatrtc, &sysctl_lasat_rtc},
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "rtc",
+  .data  = &rtctmp,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dolasatrtc,
+  .strategy = &sysctl_lasat_rtc
+ },
 #endif
- {LASAT_NAMESTR, "namestr", &lasat_board_info.li_namestr,
sizeof(lasat_board_info.li_namestr),
-  0444, NULL, &proc_dostring, &sysctl_string},
- {LASAT_TYPESTR, "typestr", &lasat_board_info.li_typestr, sizeof(lasat_board_info.li_typestr),
-  0444, NULL, &proc_dostring, &sysctl_string},
- {0}
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "namestr",
+  .data  = &lasat_board_info.li_namestr,
+  .maxlen  = sizeof(lasat_board_info.li_namestr),
+  .mode  = 0444,
+  .proc_handler =  &proc_dostring,
+  .strategy = &sysctl_string
+ },
+ {
```

```
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "typestr",
+  .data  = &lasat_board_info.li_typestr,
+  .maxlen  = sizeof(lasat_board_info.li_typestr),
+  .mode  = 0444,
+  .proc_handler = &proc_dostring,
+  .strategy = &sysctl_string
+ },
+ {}
 };

-#define CTL_LASAT 1 // CTL_ANY ???
 static ctl_table lasat_root_table[] = {
- { CTL_LASAT, "lasat", NULL, 0, 0555, lasat_table },
- { 0 }
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "lasat",
+  .mode  =  0555,
+  .child  = lasat_table
+ },
+ {}
 };

 static int __init lasat_register_sysctl(void)
--
1.4.4.1.g278f
```

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

Subject: [PATCH 33/59] sysctl: s390 move sysctl definitions to sysctl.h
Posted by ebiederm on Tue, 16 Jan 2007 16:39:38 GMT
View Forum Message <> Reply to Message

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

We need to have the the definition of all top level sysctl
directories registers in sysctl.h so we don't conflict by
accident and cause abi problems.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/s390/appldata/appldata.h | 3 +--
 arch/s390/kernel/debug.c | 1 -

```
 arch/s390/mm/cmm.c           |    4 ----
 include/linux/sysctl.h       |    7 +++++++
 4 files changed, 8 insertions(+), 7 deletions(-)

diff --git a/arch/s390/appldata/appldata.h b/arch/s390/appldata/appldata.h
index 0429481..4069b81 100644
--- a/arch/s390/appldata/appldata.h
+++ b/arch/s390/appldata/appldata.h
@@ -21,8 +21,7 @@
 #define APPLDATA_RECORD_NET_SUM_ID 0x03 /* must be < 256 !     */
 #define APPLDATA_RECORD_PROC_ID  0x04

-#define CTL_APPLDATA   2120 /* sysctl IDs, must be unique */
-#define CTL_APPLDATA_TIMER  2121
+#define CTL_APPLDATA_TIMER  2121 /* sysctl IDs, must be unique */
 #define CTL_APPLDATA_INTERVAL  2122
 #define CTL_APPLDATA_MEM 2123
 #define CTL_APPLDATA_OS  2124
diff --git a/arch/s390/kernel/debug.c b/arch/s390/kernel/debug.c
index bb57bc0..c81f8e5 100644
--- a/arch/s390/kernel/debug.c
+++ b/arch/s390/kernel/debug.c
@@ -852,7 +852,6 @@ debug_finish_entry(debug_info_t * id, debug_entry_t* active, int level,
 static int debug_stoppable=1;
 static int debug_active=1;

-#define CTL_S390DBF 5677
 #define CTL_S390DBF_STOPPABLE 5678
 #define CTL_S390DBF_ACTIVE 5679

diff --git a/arch/s390/mm/cmm.c b/arch/s390/mm/cmm.c
index 607f50e..df733d5 100644
--- a/arch/s390/mm/cmm.c
+++ b/arch/s390/mm/cmm.c
@@ -256,10 +256,6 @@ cmm_skip_blanks(char *cp, char **endp)
 }

 #ifdef CONFIG_CMM_PROC
-/* These will someday get removed. */
-#define VM_CMM_PAGES  1111
-#define VM_CMM_TIMED_PAGES 1112
-#define VM_CMM_TIMEOUT  1113

 static struct ctl_table cmm_table[];

diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 71c16b4..56d0161 100644
--- a/include/linux/sysctl.h
```

```
+++ b/include/linux/sysctl.h
@@ -73,6 +73,8 @@ enum
  CTL_SUNRPC=7249, /* sunrpc debug */
  CTL_PM=9899,  /* frv power management */
  CTL_FRV=9898,  /* frv specific sysctls */
+ CTL_S390DBF=5677, /* s390 debug */
+ CTL_APPLDATA=2120, /* s390 appldata */
 };

 /* CTL_BUS names: */
@@ -205,6 +207,11 @@ enum
  VM_PANIC_ON_OOM=33, /* panic at out-of-memory */
  VM_VDSO_ENABLED=34, /* map VDSO into new processes? */
  VM_MIN_SLAB=35,   /* Percent pages ignored by zone reclaim */
+
+ /* s390 vm cmm sysctls */
+ VM_CMM_PAGES=1111,
+ VM_CMM_TIMED_PAGES=1112,
+ VM_CMM_TIMEOUT=1113,
 };


--
1.4.4.1.g278f
```

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

Subject: [PATCH 34/59] sysctl: s390 Remove unnecessary use of insert_at_head
Posted by ebiederm on Tue, 16 Jan 2007 16:39:39 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
 arch/s390/appldata/appldata_base.c |    4 ++--
 arch/s390/kernel/debug.c           |    2 +-
 arch/s390/mm/cmm.c                 |    2 +-
 3 files changed, 4 insertions(+), 4 deletions(-)

diff --git a/arch/s390/appldata/appldata_base.c b/arch/s390/appldata/appldata_base.c
index b8c2372..cdc4109 100644
--- a/arch/s390/appldata/appldata_base.c
+++ b/arch/s390/appldata/appldata_base.c
```

```
@@ -506,7 +506,7 @@ int appldata_register_ops(struct appldata_ops *ops)

  ops->ctl_table[3].ctl_name = 0;

- ops->sysctl_header = register_sysctl_table(ops->ctl_table,1);
+ ops->sysctl_header = register_sysctl_table(ops->ctl_table,0);

  P_INFO("%s-ops registered!\n", ops->name);
  return 0;
@@ -606,7 +606,7 @@ static int __init appldata_init(void)
  /* Register cpu hotplug notifier */
  register_hotcpu_notifier(&appldata_nb);

- appldata_sysctl_header = register_sysctl_table(appldata_dir_table, 1);
+ appldata_sysctl_header = register_sysctl_table(appldata_dir_table, 0);
 #ifdef MODULE
  appldata_dir_table[0].de->owner = THIS_MODULE;
  appldata_table[0].de->owner = THIS_MODULE;
diff --git a/arch/s390/kernel/debug.c b/arch/s390/kernel/debug.c
index c81f8e5..d38cb27 100644
--- a/arch/s390/kernel/debug.c
+++ b/arch/s390/kernel/debug.c
@@ -1053,7 +1053,7 @@ __init debug_init(void)
 {
  int rc = 0;

- s390dbf_sysctl_header = register_sysctl_table(s390dbf_dir_table, 1);
+ s390dbf_sysctl_header = register_sysctl_table(s390dbf_dir_table, 0);
  down(&debug_lock);
  debug_debugfs_root_entry = debugfs_create_dir(DEBUG_DIR_ROOT,NULL);
  printk(KERN_INFO "debug: Initialization complete\n");
diff --git a/arch/s390/mm/cmm.c b/arch/s390/mm/cmm.c
index df733d5..5f83a3f 100644
--- a/arch/s390/mm/cmm.c
+++ b/arch/s390/mm/cmm.c
@@ -418,7 +418,7 @@ cmm_init (void)
  int rc = -ENOMEM;

 #ifdef CONFIG_CMM_PROC
- cmm_sysctl_header = register_sysctl_table(cmm_dir_table, 1);
+ cmm_sysctl_header = register_sysctl_table(cmm_dir_table, 0);
  if (!cmm_sysctl_header)
   goto out;
 #endif
--
1.4.4.1.g278f
```

Subject: [PATCH 35/59] sysctl: C99 convert ctl_tables in arch/powerpc/kernel/idle.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:40 GMT
View Forum Message <> Reply to Message

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

This was partially done already and there was no ABI breakage what
a relief.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/powerpc/kernel/idle.c | 11 ++++++++---
 1 files changed, 8 insertions(+), 3 deletions(-)

diff --git a/arch/powerpc/kernel/idle.c b/arch/powerpc/kernel/idle.c
index 8994af3..8b27bb1 100644
--- a/arch/powerpc/kernel/idle.c
+++ b/arch/powerpc/kernel/idle.c
@@ -110,11 +110,16 @@ static ctl_table powersave_nap_ctl_table[]={
   .mode = 0644,
   .proc_handler = &proc_dointvec,
  },
- { 0, },
+ {}
 };
 static ctl_table powersave_nap_sysctl_root[] = {
- { 1, "kernel", NULL, 0, 0755, powersave_nap_ctl_table, },
- { 0,},
+ {
+ .ctl_name = CTL_KERN,
+ .procname = "kernel",
+ .mode = 0755,
+ .child = powersave_nap_ctl_table,
+ },
+ {}
 };

 static int __init
--
1.4.4.1.g278f

_____
Containers mailing list

Subject: [PATCH 36/59] sysctl: C99 convert ctl_tables entries in arch/ppc/kernel/ppc_htab.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:41 GMT
View Forum Message <> Reply to Message

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

And make the mode of the kernel directory 0555 no one is allowed
to write to sysctl directories.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/ppc/kernel/ppc_htab.c | 11 ++++++++---
 1 files changed, 8 insertions(+), 3 deletions(-)

```
diff --git a/arch/ppc/kernel/ppc_htab.c b/arch/ppc/kernel/ppc_htab.c
index bd129d3..77b20ff 100644
--- a/arch/ppc/kernel/ppc_htab.c
+++ b/arch/ppc/kernel/ppc_htab.c
@@ -442,11 +442,16 @@ static ctl_table htab_ctl_table[]={
    .mode  = 0644,
    .proc_handler = &proc_dol2crvec,
  },
- { 0, },
+ {}
 };
 static ctl_table htab_sysctl_root[] = {
- { 1, "kernel", NULL, 0, 0755, htab_ctl_table, },
-  { 0,},
+ {
+  .ctl_name = CTL_KERN,
+  .procname = "kernel",
+  .mode  = 0555,
+  .child  = htab_ctl_table,
+ },
+ {}
 };

 static int __init
--
1.4.4.1.g278f
```

_____
Containers mailing list

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

Subject: [PATCH 37/59] sysctl: C99 convert arch/sh64/kernel/traps.c and remove ABI breakage.
Posted by ebiederm on Tue, 16 Jan 2007 16:39:42 GMT
View Forum Message <> Reply to Message

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

While doing the C99 conversion I notices that the top level sh64
directory was using the binary number for CTL_KERN. That is a
no-no so I removed the support for the sysctl binary interface
only leaving sysctl /proc support.

At least the sysctl tables were placed at the end of
the list so user space did not see this mistake.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/sh64/kernel/traps.c | 49 ++++++++++++++++++++++++++++++++++++----------
 1 files changed, 38 insertions(+), 11 deletions(-)

diff --git a/arch/sh64/kernel/traps.c b/arch/sh64/kernel/traps.c
index 224b7f5..02cca74 100644
--- a/arch/sh64/kernel/traps.c
+++ b/arch/sh64/kernel/traps.c
@@ -910,25 +910,52 @@ static int misaligned_fixup(struct pt_regs *regs)
 }

 static ctl_table unaligned_table[] = {
- {1, "kernel_reports", &kernel_mode_unaligned_fixup_count,
- sizeof(int), 0644, NULL, &proc_dointvec},
+ {
+ .ctl_name = CTL_UNNUMBERED,
+ .procname = "kernel_reports",
+ .data = &kernel_mode_unaligned_fixup_count,
+ .maxlen = sizeof(int),
+ .mode = 0644,
+ .proc_handler = &proc_dointvec
+ },
 #if defined(CONFIG_SH64_USER_MISALIGNED_FIXUP)
- {2, "user_reports", &user_mode_unaligned_fixup_count,
- sizeof(int), 0644, NULL, &proc_dointvec},
- {3, "user_enable", &user_mode_unaligned_fixup_enable,
- sizeof(int), 0644, NULL, &proc_dointvec},
+ {

```
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "user_reports",
+  .data  = &user_mode_unaligned_fixup_count,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "user_enable",
+  .data  = &user_mode_unaligned_fixup_enable,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec},
 #endif
- {0}
+ {}
 };

 static ctl_table unaligned_root[] = {
- {1, "unaligned_fixup", NULL, 0, 0555, unaligned_table},
- {0}
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "unaligned_fixup",
+  .mode  = 0555,
+  unaligned_table
+ },
+ {}
 };

 static ctl_table sh64_root[] = {
- {1, "sh64", NULL, 0, 0555, unaligned_root},
- {0}
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "sh64",
+  .mode  = 0555,
+  .child  = unaligned_root
+ },
+ {}
 };
 static struct ctl_table_header *sysctl_header;
 static int __init init_sysctl(void)
--
1.4.4.1.g278f
```

Subject: [PATCH 38/59] sysctl: x86_64 Remove unnecessary use of insert_at_head
Posted by ebiederm on Tue, 16 Jan 2007 16:39:43 GMT
View Forum Message <> Reply to Message

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

The only sysctl x86_64 provides are not provided elsewhere,
so insert_at_head is unnecessary.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/x86_64/ia32/ia32_binfmt.c | 2 +-
 arch/x86_64/mm/init.c | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)

```
diff --git a/arch/x86_64/ia32/ia32_binfmt.c b/arch/x86_64/ia32/ia32_binfmt.c
index 543ef4f..75677ad 100644
--- a/arch/x86_64/ia32/ia32_binfmt.c
+++ b/arch/x86_64/ia32/ia32_binfmt.c
@@ -408,7 +408,7 @@ static ctl_table abi_root_table2[] = {

 static __init int ia32_binfmt_init(void)
 {
- register_sysctl_table(abi_root_table2, 1);
+ register_sysctl_table(abi_root_table2, 0);
  return 0;
 }
 __initcall(ia32_binfmt_init);
diff --git a/arch/x86_64/mm/init.c b/arch/x86_64/mm/init.c
index 2968b90..65aa66c 100644
--- a/arch/x86_64/mm/init.c
+++ b/arch/x86_64/mm/init.c
@@ -724,7 +724,7 @@ static ctl_table debug_root_table2[] = {

 static __init int x8664_sysctl_init(void)
 {
- register_sysctl_table(debug_root_table2, 1);
+ register_sysctl_table(debug_root_table2, 0);
  return 0;
 }
 __initcall(x8664_sysctl_init);
--
1.4.4.1.g278f
```

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

Subject: [PATCH 39/59] sysctl: C99 convert ctl_tables in arch/x86_64/ia32/ia32_binfmt.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:44 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/x86_64/ia32/ia32_binfmt.c | 30 ++++++++++++++++++++----------
 1 files changed, 20 insertions(+), 10 deletions(-)

diff --git a/arch/x86_64/ia32/ia32_binfmt.c b/arch/x86_64/ia32/ia32_binfmt.c
index 75677ad..644b203 100644
--- a/arch/x86_64/ia32/ia32_binfmt.c
+++ b/arch/x86_64/ia32/ia32_binfmt.c
@@ -395,16 +395,26 @@ EXPORT_SYMBOL(ia32_setup_arg_pages);
 #include <linux/sysctl.h>

 static ctl_table abi_table2[] = {
- { 99, "vsyscall32", &sysctl_vsyscall32, sizeof(int), 0644, NULL,
-   proc_dointvec },
- { 0, }
-};
-
-static ctl_table abi_root_table2[] = {
- { .ctl_name = CTL_ABI, .procname = "abi", .mode = 0555,
-   .child = abi_table2 },
- { 0 },
-};
+ {
+ .ctl_name = 99,
+ .procname = "vsyscall32",
+ .data = &sysctl_vsyscall32,
+ .maxlen = sizeof(int),
+ .mode = 0644,
+ .proc_handler = proc_dointvec
+ },
+ {}
+};
+

```
+static ctl_table abi_root_table2[] = {
+ {
+  .ctl_name = CTL_ABI,
+  .procname = "abi",
+  .mode = 0555,
+  .child = abi_table2
+ },
+ {}
+};

 static __init int ia32_binfmt_init(void)
 {
--
1.4.4.1.g278f
```

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

Subject: [PATCH 40/59] sysctl: C99 convert ctl_tables in arch/x86_64/kernel/vsyscall.c

Posted by ebiederm on Tue, 16 Jan 2007 16:39:45 GMT

View Forum Message <> Reply to Message

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

Basically everything was done but I removed all element
initializers from the trailing entries to make it clear
the entire last entry should be zero filled.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 arch/x86_64/kernel/vsyscall.c |    4 ++--
 1 files changed, 2 insertions(+), 2 deletions(-)

diff --git a/arch/x86_64/kernel/vsyscall.c b/arch/x86_64/kernel/vsyscall.c
index 2433d6f..c0e2b48 100644
--- a/arch/x86_64/kernel/vsyscall.c
+++ b/arch/x86_64/kernel/vsyscall.c
@@ -235,13 +235,13 @@ static ctl_table kernel_table2[] = {
    .data = &sysctl_vsyscall, .maxlen = sizeof(int), .mode = 0644,
    .strategy = vsyscall_sysctl_nostrat,
    .proc_handler = vsyscall_sysctl_change },
- { 0, }
+ {}
 };
```

```
 static ctl_table kernel_root_table2[] = {
  { .ctl_name = CTL_KERN, .procname = "kernel", .mode = 0555,
    .child = kernel_table2 },
- { 0 },
+ {}
 };

 #endif
--
1.4.4.1.g278f
```

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

Subject: [PATCH 41/59] sysctl: C99 convert ctl_tables in arch/x86_64/mm/init.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:46 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 arch/x86_64/mm/init.c | 22 ++++++++++++++++------
 1 files changed, 16 insertions(+), 6 deletions(-)

```
diff --git a/arch/x86_64/mm/init.c b/arch/x86_64/mm/init.c
index 65aa66c..a04535d 100644
--- a/arch/x86_64/mm/init.c
+++ b/arch/x86_64/mm/init.c
@@ -711,15 +711,25 @@ int kern_addr_valid(unsigned long addr)
 extern int exception_trace, page_fault_trace;

 static ctl_table debug_table2[] = {
- { 99, "exception-trace", &exception_trace, sizeof(int), 0644, NULL,
-   proc_dointvec },
- { 0, }
+ {
+  .ctl_name = 99,
+  .procname = "exception-trace",
+  .data  = &exception_trace,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = proc_dointvec
+ },
```

```
+ {}
 };

 static ctl_table debug_root_table2[] = {
- { .ctl_name = CTL_DEBUG, .procname = "debug", .mode = 0555,
-    .child = debug_table2 },
- { 0 },
+ {
+  .ctl_name = CTL_DEBUG,
+  .procname = "debug",
+  .mode = 0555,
+  .child = debug_table2
+ },
+ {}
 };

 static __init int x8664_sysctl_init(void)
--
1.4.4.1.g278f
```

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

Subject: [PATCH 42/59] sysctl: Remove sys_sysctl support from the hpet timer driver.
Posted by ebiederm on Tue, 16 Jan 2007 16:39:47 GMT
View Forum Message <> Reply to Message

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

In the binary sysctl interface the hpet driver was claiming to
be the cdrom driver. This is a no-no so remove support for the
binary interface.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/char/hpet.c | 4 ++--
 1 files changed, 2 insertions(+), 2 deletions(-)

diff --git a/drivers/char/hpet.c b/drivers/char/hpet.c
index 20dc3be..81be1db 100644
--- a/drivers/char/hpet.c
+++ b/drivers/char/hpet.c
@@ -703,7 +703,7 @@ int hpet_control(struct hpet_task *tp, unsigned int cmd, unsigned long
arg)

```
 static ctl_table hpet_table[] = {
  {
-  .ctl_name = 1,
+  .ctl_name = CTL_UNNUMBERED,
    .procname = "max-user-freq",
    .data = &hpet_max_freq,
    .maxlen = sizeof(int),
@@ -715,7 +715,7 @@ static ctl_table hpet_table[] = {

 static ctl_table hpet_root[] = {
  {
-  .ctl_name = 1,
+  .ctl_name = CTL_UNNUMBERED,
    .procname = "hpet",
    .maxlen = 0,
    .mode = 0555,
--
1.4.4.1.g278f
```

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

Subject: [PATCH 43/59] sysctl: Remove sys_sysctl support from drivers/char/rtc.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:48 GMT
View Forum Message <> Reply to Message

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

The real time clock driver was using the binary number reserved
for cdroms in the sysctl binary number interface, which is a no-no.
So since the sysctl binary interface is wrong remove it.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/char/rtc.c | 6 ++----
 1 files changed, 2 insertions(+), 4 deletions(-)

```
diff --git a/drivers/char/rtc.c b/drivers/char/rtc.c
index 664f36c..df11289 100644
--- a/drivers/char/rtc.c
+++ b/drivers/char/rtc.c
@@ -282,7 +282,7 @@ irqreturn_t rtc_interrupt(int irq, void *dev_id)
  */
 static ctl_table rtc_table[] = {
```

```
 {
-  .ctl_name = 1,
+  .ctl_name = CTL_UNNUMBERED,
   .procname = "max-user-freq",
   .data  = &rtc_max_user_freq,
   .maxlen  = sizeof(int),
@@ -294,9 +294,8 @@ static ctl_table rtc_table[] = {

 static ctl_table rtc_root[] = {
  {
-  .ctl_name = 1,
+  .ctl_name = CTL_UNNUMBERED,
   .procname = "rtc",
-  .maxlen  = 0,
   .mode  = 0555,
   .child  = rtc_table,
  },
@@ -307,7 +306,6 @@ static ctl_table dev_root[] = {
  {
   .ctl_name = CTL_DEV,
   .procname = "dev",
-  .maxlen  = 0,
   .mode  = 0555,
   .child  = rtc_root,
  },
--
1.4.4.1.g278f
```

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

Subject: [PATCH 44/59] sysctl: Register the sysctl number used by the arlan driver.
Posted by ebiederm on Tue, 16 Jan 2007 16:39:49 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/net/wireless/arlan-proc.c | 2 +-
 include/linux/sysctl.h | 1 +
 2 files changed, 2 insertions(+), 1 deletions(-)

diff --git a/drivers/net/wireless/arlan-proc.c b/drivers/net/wireless/arlan-proc.c
index 5fa9854..20499a6 100644

```
--- a/drivers/net/wireless/arlan-proc.c
+++ b/drivers/net/wireless/arlan-proc.c
@@ -1216,7 +1216,7 @@ static ctl_table arlan_table[MAX_ARLANS + 1] =
 static ctl_table arlan_root_table[] =
 {
  {
-  .ctl_name = 254,
+  .ctl_name = CTL_ARLAN,
    .procname = "arlan",
    .maxlen  = 0,
    .mode  = 0555,
diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 56d0161..f4ba72e 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -70,6 +70,7 @@ enum
  CTL_BUS=8,  /* Busses */
  CTL_ABI=9,  /* Binary emulation */
  CTL_CPU=10,  /* CPU stuff (speed scaling, etc) */
+ CTL_ARLAN=254,  /* arlan wireless driver */
  CTL_SUNRPC=7249, /* sunrpc debug */
  CTL_PM=9899,  /* frv power management */
  CTL_FRV=9898,  /* frv specific sysctls */
--
1.4.4.1.g278f
```

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

Subject: [PATCH 45/59] sysctl: C99 convert ctl_tables in drivers/parport/procfs.c
Posted by ebiederm on Tue, 16 Jan 2007 16:39:50 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/parport/procfs.c | 264 ++++++

```
  return copy_to_user(result, buffer, len) ? -EFAULT : 0;
 }

-#define PARPORT_PORT_DIR(child) { 0, NULL, NULL, 0, 0555, child }
-#define PARPORT_PARPORT_DIR(child) { DEV_PARPORT, "parport", \
-                                     NULL, 0, 0555, child }
-#define PARPORT_DEV_DIR(child) { CTL_DEV, "dev", NULL, 0, 0555, child }
-#define PARPORT_DEVICES_ROOT_DIR  { DEV_PARPORT_DEVICES, "devices", \
-                                    NULL, 0, 0555, NULL }
+#define PARPORT_PORT_DIR(CHILD) { .ctl_name = 0, .procname = NULL, .mode = 0555,
.child = CHILD }
+#define PARPORT_PARPORT_DIR(CHILD) { .ctl_name = DEV_PARPORT, .procname =
"parport", \
+                                     .mode = 0555, .child = CHILD }
+#define PARPORT_DEV_DIR(CHILD) { .ctl_name = CTL_DEV, .procname = "dev", .mode =
0555, .child = CHILD }
+#define PARPORT_DEVICES_ROOT_DIR  { .ctl_name = DEV_PARPORT_DEVICES,
.procname = "devices", \
+                                    .mode = 0555, .child = NULL }

 static const unsigned long parport_min_timeslice_value =
 PARPORT_MIN_TIMESLICE_VALUE;
@@ -263,50 +263,118 @@ struct parport_sysctl_table {
 };

 static const struct parport_sysctl_table parport_sysctl_template = {
- NULL,
+ .sysctl_header = NULL,
         {
-  { DEV_PARPORT_SPINTIME, "spintime",
-    NULL, sizeof(int), 0644, NULL,
-    &proc_dointvec_minmax, NULL, NULL,
-    (void*) &parport_min_spintime_value,
-    (void*) &parport_max_spintime_value },
-  { DEV_PARPORT_BASE_ADDR, "base-addr",
-    NULL, 0, 0444, NULL,
-    &do_hardware_base_addr },
-  { DEV_PARPORT_IRQ, "irq",
-    NULL, 0, 0444, NULL,
-    &do_hardware_irq },
-  { DEV_PARPORT_DMA, "dma",
-    NULL, 0, 0444, NULL,
-    &do_hardware_dma },
-  { DEV_PARPORT_MODES, "modes",
-    NULL, 0, 0444, NULL,
-    &do_hardware_modes },
+  {
+   .ctl_name = DEV_PARPORT_SPINTIME,
```

```
+   .procname = "spintime",
+   .data  = NULL,
+   .maxlen  = sizeof(int),
+   .mode  = 0644,
+   .proc_handler = &proc_dointvec_minmax,
+   .extra1  = (void*) &parport_min_spintime_value,
+   .extra2  = (void*) &parport_max_spintime_value
+  },
+  {
+   .ctl_name = DEV_PARPORT_BASE_ADDR,
+   .procname = "base-addr",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_hardware_base_addr
+  },
+  {
+   .ctl_name = DEV_PARPORT_IRQ,
+   .procname = "irq",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_hardware_irq
+  },
+  {
+   .ctl_name = DEV_PARPORT_DMA,
+   .procname = "dma",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_hardware_dma
+  },
+  {
+   .ctl_name = DEV_PARPORT_MODES,
+   .procname = "modes",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_hardware_modes
+  },
    PARPORT_DEVICES_ROOT_DIR,
  #ifdef CONFIG_PARPORT_1284
-  { DEV_PARPORT_AUTOPROBE, "autoprobe",
-    NULL, 0, 0444, NULL,
-    &do_autoprobe },
-  { DEV_PARPORT_AUTOPROBE + 1, "autoprobe0",
-   NULL, 0, 0444, NULL,
-   &do_autoprobe },
```

```
- { DEV_PARPORT_AUTOPROBE + 2, "autoprobe1",
-   NULL, 0, 0444, NULL,
-   &do_autoprobe },
- { DEV_PARPORT_AUTOPROBE + 3, "autoprobe2",
-   NULL, 0, 0444, NULL,
-   &do_autoprobe },
- { DEV_PARPORT_AUTOPROBE + 4, "autoprobe3",
-   NULL, 0, 0444, NULL,
-   &do_autoprobe },
+ {
+   .ctl_name = DEV_PARPORT_AUTOPROBE,
+   .procname = "autoprobe",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_autoprobe
+ },
+ {
+   .ctl_name = DEV_PARPORT_AUTOPROBE + 1,
+   .procname = "autoprobe0",
+   .data  = NULL,
+   .maxlen  = 0,
+   .maxlen  = 0444,
+   .proc_handler =  &do_autoprobe
+ },
+ {
+   .ctl_name = DEV_PARPORT_AUTOPROBE + 2,
+   .procname = "autoprobe1",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_autoprobe
+ },
+ {
+   .ctl_name = DEV_PARPORT_AUTOPROBE + 3,
+   .procname = "autoprobe2",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_autoprobe
+ },
+ {
+   .ctl_name = DEV_PARPORT_AUTOPROBE + 4,
+   .procname = "autoprobe3",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0444,
+   .proc_handler = &do_autoprobe
```

```
+  },
 #endif /* IEEE 1284 support */
-  {0}
+  {}
   },
- { {DEV_PARPORT_DEVICES_ACTIVE, "active", NULL, 0, 0444, NULL,
-    &do_active_device }, {0}},
- { PARPORT_PORT_DIR(NULL), {0}},
- { PARPORT_PARPORT_DIR(NULL), {0}},
- { PARPORT_DEV_DIR(NULL), {0}}
+ {
+  {
+    .ctl_name = DEV_PARPORT_DEVICES_ACTIVE,
+    .procname = "active",
+    .data  = NULL,
+    .maxlen  = 0,
+    .mode  = 0444,
+    .proc_handler = &do_active_device
+  },
+  {}
+ },
+ {
+  PARPORT_PORT_DIR(NULL),
+  {}
+ },
+ {
+  PARPORT_PARPORT_DIR(NULL),
+  {}
+ },
+ {
+  PARPORT_DEV_DIR(NULL),
+  {}
+ }
 };

 struct parport_device_sysctl_table
@@ -322,19 +390,46 @@ struct parport_device_sysctl_table

 static const struct parport_device_sysctl_table
 parport_device_sysctl_template = {
- NULL,
+ .sysctl_header = NULL,
+ {
+  {
+    .ctl_name  = DEV_PARPORT_DEVICE_TIMESLICE,
+    .procname  = "timeslice",
+    .data  = NULL,
+    .maxlen  = sizeof(int),
```

```
+   .mode  = 0644,
+   .proc_handler = &proc_doulongvec_ms_jiffies_minmax,
+   .extra1  = (void*) &parport_min_timeslice_value,
+   .extra2  = (void*) &parport_max_timeslice_value
+  },
+ },
+ {
+  {
+   .ctl_name = 0,
+   .procname = NULL,
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0555,
+   .child  = NULL
+  },
+  {}
+ },
   {
-  { DEV_PARPORT_DEVICE_TIMESLICE, "timeslice",
-    NULL, sizeof(int), 0644, NULL,
-    &proc_doulongvec_ms_jiffies_minmax, NULL, NULL,
-    (void*) &parport_min_timeslice_value,
-    (void*) &parport_max_timeslice_value },
+  PARPORT_DEVICES_ROOT_DIR,
+  {}
+ },
+ {
+  PARPORT_PORT_DIR(NULL),
+  {}
   },
- { {0, NULL, NULL, 0, 0555, NULL}, {0}},
- { PARPORT_DEVICES_ROOT_DIR, {0}},
- { PARPORT_PORT_DIR(NULL), {0}},
- { PARPORT_PARPORT_DIR(NULL), {0}},
- { PARPORT_DEV_DIR(NULL), {0}}
+ {
+  PARPORT_PARPORT_DIR(NULL),
+  {}
+ },
+ {
+  PARPORT_DEV_DIR(NULL),
+  {}
+ }
 };

 struct parport_default_sysctl_table
@@ -351,28 +446,47 @@ extern int parport_default_spintime;
```

```
 static struct parport_default_sysctl_table
 parport_default_sysctl_table = {
- NULL,
+ .sysctl_header = NULL,
+ {
+  {
+   .ctl_name = DEV_PARPORT_DEFAULT_TIMESLICE,
+   .procname = "timeslice",
+   .data  = &parport_default_timeslice,
+   .maxlen  = sizeof(parport_default_timeslice),
+   .mode  = 0644,
+   .proc_handler = &proc_doulongvec_ms_jiffies_minmax,
+   .extra1  = (void*) &parport_min_timeslice_value,
+   .extra2  = (void*) &parport_max_timeslice_value
+  },
+  {
+   .ctl_name = DEV_PARPORT_DEFAULT_SPINTIME,
+   .procname = "spintime",
+   .data  = &parport_default_spintime,
+   .maxlen  = sizeof(parport_default_spintime),
+   .mode  = 0644,
+   .proc_handler = &proc_dointvec_minmax,
+   .extra1  = (void*) &parport_min_spintime_value,
+   .extra2  = (void*) &parport_max_spintime_value
+  },
+  {}
+ },
   {
-  { DEV_PARPORT_DEFAULT_TIMESLICE, "timeslice",
-    &parport_default_timeslice,
-    sizeof(parport_default_timeslice), 0644, NULL,
-    &proc_doulongvec_ms_jiffies_minmax, NULL, NULL,
-    (void*) &parport_min_timeslice_value,
-    (void*) &parport_max_timeslice_value },
-  { DEV_PARPORT_DEFAULT_SPINTIME, "spintime",
-    &parport_default_spintime,
-    sizeof(parport_default_spintime), 0644, NULL,
-    &proc_dointvec_minmax, NULL, NULL,
-    (void*) &parport_min_spintime_value,
-    (void*) &parport_max_spintime_value },
-  {0}
+  {
+   .ctl_name = DEV_PARPORT_DEFAULT,
+   .procname = "default",
+   .mode  = 0555,
+   .child  = parport_default_sysctl_table.vars
+  },
+  {}
```

```
  },
- { { DEV_PARPORT_DEFAULT, "default", NULL, 0, 0555,
-     parport_default_sysctl_table.vars },{0}},
  {
- PARPORT_PARPORT_DIR(parport_default_sysctl_table.default_dir),
- {0}},
- { PARPORT_DEV_DIR(parport_default_sysctl_table.parport_dir), {0}}
+  PARPORT_PARPORT_DIR(parport_default_sysctl_table.default_dir),
+  {}
+ },
+ {
+  PARPORT_DEV_DIR(parport_default_sysctl_table.parport_dir),
+  {}
+ }
 };


--
1.4.4.1.g278f
```

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

Subject: [PATCH 46/59] sysctl: C99 convert coda ctl_tables and remove binary sysctls.
Posted by ebiederm on Tue, 16 Jan 2007 16:39:51 GMT
View Forum Message <> Reply to Message

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

Will converting the coda sysctl initializers I discovered that
it is yet another user of sysctl that was stomping CTL_KERN.
So off with it's sys_sysctl support since it wasn't done
in a supportable way.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 fs/coda/sysctl.c | 58 ++---------
 1 files changed, 48 insertions(+), 10 deletions(-)

diff --git a/fs/coda/sysctl.c b/fs/coda/sysctl.c
index 1c82e9a..df682e2 100644
--- a/fs/coda/sysctl.c
+++ b/fs/coda/sysctl.c
@@ -32,8 +32,6 @@

```
 static struct ctl_table_header *fs_table_header;

-#define FS_CODA         1       /* Coda file system */
-
 #define CODA_TIMEOUT    3       /* timeout on upcalls to become intrble */
 #define CODA_HARD       5       /* mount type "hard" or "soft" */
 #define CODA_VFS   6       /* vfs statistics */
@@ -184,17 +182,57 @@ static int coda_cache_inv_stats_get_info( char * buffer, char ** start,
 }

 static ctl_table coda_table[] = {
-  {CODA_TIMEOUT, "timeout", &coda_timeout, sizeof(int), 0644, NULL, &proc_dointvec},
-  {CODA_HARD, "hard", &coda_hard, sizeof(int), 0644, NULL, &proc_dointvec},
-  {CODA_VFS, "vfs_stats", NULL, 0, 0644, NULL, &do_reset_coda_vfs_stats},
-  {CODA_CACHE_INV, "cache_inv_stats", NULL, 0, 0644, NULL,
&do_reset_coda_cache_inv_stats},
-  {CODA_FAKE_STATFS, "fake_statfs", &coda_fake_statfs, sizeof(int), 0600, NULL,
&proc_dointvec},
- { 0 }
+ {
+   .ctl_name = CTL_UNNUMBERED,
+   .procname = "timeout",
+   .data  = &coda_timeout,
+   .maxlen  = sizeof(int),
+   .mode  = 0644,
+   .proc_handler = &proc_dointvec
+ },
+ {
+   .ctl_name = CTL_UNNUMBERED,
+   .procname = "hard",
+   .data  = &coda_hard,
+   .maxlen  = sizeof(int),
+   .mode  = 0644,
+   .proc_handler = &proc_dointvec
+ },
+ {
+   .ctl_name = CTL_UNNUMBERED,
+   .procname = "vfs_stats",
+   .data  = NULL,
+   .maxlen  = 0,
+   .mode  = 0644,
+   .proc_handler = &do_reset_coda_vfs_stats
+ },
+ {
+   .ctl_name = CTL_UNNUMBERED,
+   .procname = "cache_inv_stats",
+   .data  = NULL,
```

```
+  .maxlen  = 0,
+  .mode  = 0644,
+  .proc_handler = &do_reset_coda_cache_inv_stats
+ },
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "fake_statfs",
+  .data  = &coda_fake_statfs,
+  .maxlen  = sizeof(int),
+  .mode  = 0600,
+  .proc_handler = &proc_dointvec
+ },
+ {}
 };

 static ctl_table fs_table[] = {
-       {FS_CODA, "coda",    NULL, 0, 0555, coda_table},
-       {0}
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "coda",
+  .mode  = 0555,
+  .child  = coda_table
+ },
+ {}
 };


--
1.4.4.1.g278f

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

Subject: [PATCH 47/59] sysctl: C99 convert ctl_tables in NTFS and remove sys_sysctl support

Posted by ebiederm on Tue, 16 Jan 2007 16:39:52 GMT

View Forum Message <> Reply to Message

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

Putting ntfs-debug under FS_NRINODE was not a kosher thing to do
so don't give it any binary number.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 fs/ntfs/sysctl.c |   24 ++++++++++++++++++++--------
 1 files changed, 16 insertions(+), 8 deletions(-)

diff --git a/fs/ntfs/sysctl.c b/fs/ntfs/sysctl.c
index 1c23138..bc217de 100644
--- a/fs/ntfs/sysctl.c
+++ b/fs/ntfs/sysctl.c
@@ -33,20 +33,28 @@
 #include "sysctl.h"
 #include "debug.h"

-#define FS_NTFS 1
-
 /* Definition of the ntfs sysctl. */
 static ctl_table ntfs_sysctls[] = {
- { FS_NTFS, "ntfs-debug",  /* Binary and text IDs. */
-   &debug_msgs,sizeof(debug_msgs), /* Data pointer and size. */
-   0644, NULL, &proc_dointvec },  /* Mode, child, proc handler. */
- { 0 }
+ {
+  .ctl_name = CTL_UNUMBERED, /* Binary and text IDs. */
+  .procname = "ntfs-debug",
+  .data  = &debug_msgs,  /* Data pointer and size. */
+  .maxlen  = sizeof(debug_msgs),
+  .mode  = 0644,   /* Mode, proc handler. */
+  .proc_handler = &proc_dointvec
+ },
+ {}
 };

 /* Define the parent directory /proc/sys/fs. */
 static ctl_table sysctls_root[] = {
- { CTL_FS, "fs", NULL, 0, 0555, ntfs_sysctls },
- { 0 }
+ {
+  .ctl_name = CTL_FS,
+  .procname = "fs",
+  .mode  = 0555,
+  .child  = ntfs_sysctls
+ },
+ {}
 };

 /* Storage for the sysctls header. */
--
1.4.4.1.g278f
```

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

Subject: [PATCH 48/59] sysctl: Register the ocfs2 sysctl numbers
Posted by ebiederm on Tue, 16 Jan 2007 16:39:53 GMT
View Forum Message <> Reply to Message

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

ocfs2 was did not have the binary number it uses under CTL_FS
registered in sysctl.h. Register it to avoid future conflicts,
and change the name of the definition to be in line with the
rest of the sysctl numbers.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
fs/ocfs2/cluster/nodemanager.c | 4 ++--
fs/ocfs2/cluster/nodemanager.h | 3 +--
include/linux/sysctl.h | 1 +
3 files changed, 4 insertions(+), 4 deletions(-)

```
diff --git a/fs/ocfs2/cluster/nodemanager.c b/fs/ocfs2/cluster/nodemanager.c
index b17333a..df763c7 100644
--- a/fs/ocfs2/cluster/nodemanager.c
+++ b/fs/ocfs2/cluster/nodemanager.c
@@ -55,7 +55,7 @@ static ctl_table ocfs2_nm_table[] = {

 static ctl_table ocfs2_mod_table[] = {
  {
-  .ctl_name = KERN_OCFS2_NM,
+  .ctl_name = FS_OCFS2_NM,
    .procname = "nm",
    .data  = NULL,
    .maxlen  = 0,
@@ -67,7 +67,7 @@ static ctl_table ocfs2_mod_table[] = {

 static ctl_table ocfs2_kern_table[] = {
  {
-  .ctl_name = KERN_OCFS2,
+  .ctl_name = FS_OCFS2,
    .procname = "ocfs2",
    .data  = NULL,
    .maxlen  = 0,
diff --git a/fs/ocfs2/cluster/nodemanager.h b/fs/ocfs2/cluster/nodemanager.h
index 8fb23ca..0705221 100644
```

```
--- a/fs/ocfs2/cluster/nodemanager.h
+++ b/fs/ocfs2/cluster/nodemanager.h
@@ -33,8 +33,7 @@
 #include <linux/configfs.h>
 #include <linux/rbtree.h>

-#define KERN_OCFS2  988
-#define KERN_OCFS2_NM  1
+#define FS_OCFS2_NM  1

 const char *o2nm_get_hb_ctl_path(void);

diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index f4ba72e..63e1bac 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -813,6 +813,7 @@ enum
  FS_AIO_NR=18, /* current system-wide number of aio requests */
  FS_AIO_MAX_NR=19, /* system-wide maximum number of aio requests */
  FS_INOTIFY=20, /* inotify submenu */
+ FS_OCFS2=988, /* ocfs2 */
 };

 /* /proc/sys/fs/quota/ */
--
1.4.4.1.g278f
```

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

Subject: [PATCH 49/59] sysctl: Move init_irq_proc into init/main where it belongs
Posted by ebiederm on Tue, 16 Jan 2007 16:39:54 GMT

View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
---
 init/main.c     |    3 +++
 kernel/sysctl.c |    3 ---
 2 files changed, 3 insertions(+), 3 deletions(-)

diff --git a/init/main.c b/init/main.c
index 8b4a7d7..8af5c6e 100644
--- a/init/main.c
```

```
+++ b/init/main.c
@@ -691,6 +691,9 @@ static void __init do_basic_setup(void)
 #ifdef CONFIG_SYSCTL
  sysctl_init();
 #endif
+#ifdef CONFIG_PROC_FS
+ init_irq_proc();
+#endif

  do_initcalls();
 }
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index 600b333..7420761 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -1172,8 +1172,6 @@ static ctl_table dev_table[] = {
  { .ctl_name = 0 }
 };

-extern void init_irq_proc (void);
-
 static DEFINE_SPINLOCK(sysctl_lock);

 /* called under sysctl_lock */
@@ -1219,7 +1217,6 @@ void __init sysctl_init(void)
 {
 #ifdef CONFIG_PROC_SYSCTL
  register_proc_table(root_table, proc_sys_root, &root_table_header);
- init_irq_proc();
 #endif
 }

--
1.4.4.1.g278f
```

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

Subject: [PATCH 50/59] sysctl: Move utsname sysctls to their own file
Posted by ebiederm on Tue, 16 Jan 2007 16:39:55 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 utsname logic to together it makes
the code a little more readable.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 kernel/Makefile | 1 +
 kernel/sysctl.c | 115 ----------------------------------
 kernel/utsname_sysctl.c | 146 ++++++

```
-#ifdef CONFIG_UTS_NS
- struct uts_namespace *uts_ns = current->nsproxy->uts_ns;
- which = (which - (char *)&init_uts_ns) + (char *)uts_ns;
-#endif
- if (!write)
-  down_read(&uts_sem);
- else
-  down_write(&uts_sem);
- return which;
-}
-
-static void put_uts(ctl_table *table, int write, void *which)
-{
- if (!write)
-  up_read(&uts_sem);
- else
-  up_write(&uts_sem);
-}

 #ifdef CONFIG_SYSVIPC
 static void *get_ipc(ctl_table *table, int write)
@@ -275,51 +247,6 @@ static ctl_table root_table[] = {

 static ctl_table kern_table[] = {
  {
-  .ctl_name = KERN_OSTYPE,
-  .procname = "ostype",
-  .data  = init_uts_ns.name.sysname,
-  .maxlen  = sizeof(init_uts_ns.name.sysname),
-  .mode  = 0444,
-  .proc_handler = &proc_do_uts_string,
-  .strategy = &sysctl_uts_string,
- },
- {
-  .ctl_name = KERN_OSRELEASE,
-  .procname = "osrelease",
-  .data  = init_uts_ns.name.release,
-  .maxlen  = sizeof(init_uts_ns.name.release),
-  .mode  = 0444,
-  .proc_handler = &proc_do_uts_string,
-  .strategy = &sysctl_uts_string,
- },
- {
-  .ctl_name = KERN_VERSION,
-  .procname = "version",
-  .data  = init_uts_ns.name.version,
-  .maxlen  = sizeof(init_uts_ns.name.version),
-  .mode  = 0444,
```

```
-  .proc_handler = &proc_do_uts_string,
-  .strategy = &sysctl_uts_string,
- },
- {
-  .ctl_name = KERN_NODENAME,
-  .procname = "hostname",
-  .data  = init_uts_ns.name.nodename,
-  .maxlen  = sizeof(init_uts_ns.name.nodename),
-  .mode  = 0644,
-  .proc_handler = &proc_do_uts_string,
-  .strategy = &sysctl_uts_string,
- },
- {
-  .ctl_name = KERN_DOMAINNAME,
-  .procname = "domainname",
-  .data  = init_uts_ns.name.domainname,
-  .maxlen  = sizeof(init_uts_ns.name.domainname),
-  .mode  = 0644,
-  .proc_handler = &proc_do_uts_string,
-  .strategy = &sysctl_uts_string,
- },
- {
   .ctl_name = KERN_PANIC,
   .procname = "panic",
   .data  = &panic_timeout,
@@ -1746,21 +1673,6 @@ int proc_dostring(ctl_table *table, int write, struct file *filp,
         buffer, lenp, ppos);
 }

-/*
- * Special case of dostring for the UTS structure. This has locks
- * to observe. Should this be in kernel/sys.c ????
- */
-
-static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
-    void __user *buffer, size_t *lenp, loff_t *ppos)
-{
- int r;
- void *which;
- which = get_uts(table, write);
- r = _proc_do_string(which, table->maxlen,write,filp,buffer,lenp, ppos);
- put_uts(table, write, which);
- return r;
-}

 static int do_proc_dointvec_conv(int *negp, unsigned long *lvalp,
      int *valp,
@@ -2379,12 +2291,6 @@ int proc_dostring(ctl_table *table, int write, struct file *filp,
```

```
  return -ENOSYS;
 }

-static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
-  void __user *buffer, size_t *lenp, loff_t *ppos)
-{
- 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)
@@ -2602,21 +2508,6 @@ int sysctl_ms_jiffies(ctl_table *table, int __user *name, int nlen,
 }


-/* The generic string strategy routine: */
-static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
-    void __user *oldval, size_t __user *oldlenp,
-    void __user *newval, size_t newlen)
-{
- struct ctl_table uts_table;
- int r, write;
- write = newval && newlen;
- memcpy(&uts_table, table, sizeof(uts_table));
- uts_table.data = get_uts(table, write);
- r = sysctl_string(&uts_table, name, nlen,
-  oldval, oldlenp, newval, newlen);
- put_uts(table, write, uts_table.data);
- return r;
-}

 #ifdef CONFIG_SYSVIPC
 /* The generic sysctl ipc data routine. */
@@ -2723,12 +2614,6 @@ int sysctl_ms_jiffies(ctl_table *table, int __user *name, int nlen,
  return -ENOSYS;
 }

-static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
-    void __user *oldval, size_t __user *oldlenp,
-    void __user *newval, size_t newlen)
-{
- 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)
diff --git a/kernel/utsname_sysctl.c b/kernel/utsname_sysctl.c
```

```
new file mode 100644
index 0000000..324aa13
--- /dev/null
+++ b/kernel/utsname_sysctl.c
@@ -0,0 +1,146 @@
+/*
+ *  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/uts.h>
+#include <linux/utsname.h>
+#include <linux/version.h>
+#include <linux/sysctl.h>
+
+static void *get_uts(ctl_table *table, int write)
+{
+ char *which = table->data;
+#ifdef CONFIG_UTS_NS
+ struct uts_namespace *uts_ns = current->nsproxy->uts_ns;
+ which = (which - (char *)&init_uts_ns) + (char *)uts_ns;
+#endif
+ if (!write)
+  down_read(&uts_sem);
+ else
+  down_write(&uts_sem);
+ return which;
+}
+
+static void put_uts(ctl_table *table, int write, void *which)
+{
+ if (!write)
+  up_read(&uts_sem);
+ else
+  up_write(&uts_sem);
+}
+
+#ifdef CONFIG_PROC_FS
+/*
+ * Special case of dostring for the UTS structure. This has locks
+ * to observe. Should this be in kernel/sys.c ????
```

```
+ */
+static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
+    void __user *buffer, size_t *lenp, loff_t *ppos)
+{
+ struct ctl_table uts_table;
+ int r;
+ memcpy(&uts_table, table, sizeof(uts_table));
+ uts_table.data = get_uts(table, write);
+ r = proc_dostring(&uts_table,write,filp,buffer,lenp, ppos);
+ put_uts(table, write, uts_table.data);
+ return r;
+}
+#else
+#define proc_do_uts_string NULL
+#endif
+
+
+#ifdef CONFIG_SYSCTL_SYSCALL
+/* The generic string strategy routine: */
+static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
+    void __user *oldval, size_t __user *oldlenp,
+    void __user *newval, size_t newlen)
+{
+ struct ctl_table uts_table;
+ int r, write;
+ write = newval && newlen;
+ memcpy(&uts_table, table, sizeof(uts_table));
+ uts_table.data = get_uts(table, write);
+ r = sysctl_string(&uts_table, name, nlen,
+  oldval, oldlenp, newval, newlen);
+ put_uts(table, write, uts_table.data);
+ return r;
+}
+#else
+#define sysctl_uts_string NULL
+#endif
+
+static struct ctl_table uts_kern_table[] = {
+ {
+  .ctl_name = KERN_OSTYPE,
+  .procname = "ostype",
+  .data  = init_uts_ns.name.sysname,
+  .maxlen  = sizeof(init_uts_ns.name.sysname),
+  .mode  = 0444,
+  .proc_handler = proc_do_uts_string,
+  .strategy = sysctl_uts_string,
+ },
+ {
```

```
+  .ctl_name = KERN_OSRELEASE,
+  .procname = "osrelease",
+  .data  = init_uts_ns.name.release,
+  .maxlen  = sizeof(init_uts_ns.name.release),
+  .mode  = 0444,
+  .proc_handler = proc_do_uts_string,
+  .strategy = sysctl_uts_string,
+ },
+ {
+  .ctl_name = KERN_VERSION,
+  .procname = "version",
+  .data  = init_uts_ns.name.version,
+  .maxlen  = sizeof(init_uts_ns.name.version),
+  .mode  = 0444,
+  .proc_handler = proc_do_uts_string,
+  .strategy = sysctl_uts_string,
+ },
+ {
+  .ctl_name = KERN_NODENAME,
+  .procname = "hostname",
+  .data  = init_uts_ns.name.nodename,
+  .maxlen  = sizeof(init_uts_ns.name.nodename),
+  .mode  = 0644,
+  .proc_handler = proc_do_uts_string,
+  .strategy = sysctl_uts_string,
+ },
+ {
+  .ctl_name = KERN_DOMAINNAME,
+  .procname = "domainname",
+  .data  = init_uts_ns.name.domainname,
+  .maxlen  = sizeof(init_uts_ns.name.domainname),
+  .mode  = 0644,
+  .proc_handler = proc_do_uts_string,
+  .strategy = sysctl_uts_string,
+ },
+ {}
+};
+
+static struct ctl_table uts_root_table[] = {
+ {
+  .ctl_name = CTL_KERN,
+  .procname = "kernel",
+  .mode  = 0555,
+  .child  = uts_kern_table,
+ },
+ {}
+};
+
```

```
+static int __init utsname_sysctl_init(void)
+{
+ register_sysctl_table(uts_root_table, 0);
+ return 0;
+}
+
+__initcall(utsname_sysctl_init);
--
1.4.4.1.g278f
```

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

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

Subject: [PATCH 52/59] sysctl: Create sys/fs/binfmt_misc as an ordinary sysctl entry

Posted by ebiederm on Tue, 16 Jan 2007 16:39:57 GMT
View Forum Message <> Reply to Message

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

binfmt_misc has a mount point in the middle of the sysctl and
that mount point is created as a proc_generic directory.

Doing it that way gets in the way of cleaning up the sysctl
proc support as it continues the existence of a horrible hack.
So instead simply create the directory as an ordinary sysctl
directory. At least that removes the magic special case.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 fs/proc/root.c | 4 ----
 kernel/sysctl.c | 13 +++++++++++++
 2 files changed, 13 insertions(+), 4 deletions(-)

```
diff --git a/fs/proc/root.c b/fs/proc/root.c
index 64d242b..8059e92 100644
--- a/fs/proc/root.c
+++ b/fs/proc/root.c
@@ -74,10 +74,6 @@ void __init proc_root_init(void)
 #ifdef CONFIG_SYSCTL
  proc_sys_root = proc_mkdir("sys", NULL);
 #endif
-#if defined(CONFIG_BINFMT_MISC) || defined(CONFIG_BINFMT_MISC_MODULE)
- proc_mkdir("sys/fs", NULL);
- proc_mkdir("sys/fs/binfmt_misc", NULL);
-#endif
```

```
  proc_root_fs = proc_mkdir("fs", NULL);
  proc_root_driver = proc_mkdir("driver", NULL);
  proc_mkdir("fs/nfsd", NULL); /* somewhere for the nfsd filesystem to be mounted */
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index 6e2e608..8da6647 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -142,6 +142,7 @@ static struct ctl_table_header root_table_header =
 static ctl_table kern_table[];
 static ctl_table vm_table[];
 static ctl_table fs_table[];
+static ctl_table binfmt_misc_table[];
 static ctl_table debug_table[];
 static ctl_table dev_table[];
 extern ctl_table random_table[];
@@ -1001,6 +1002,18 @@ static ctl_table fs_table[] = {
   .mode  = 0644,
   .proc_handler = &proc_dointvec,
  },
+#if defined(CONFIG_BINFMT_MISC) || defined(CONFIG_BINFMT_MISC_MODULE)
+ {
+  .ctl_name = CTL_UNNUMBERED,
+  .procname = "binfmt_misc",
+  .mode  = 0555,
+  .child  = binfmt_misc_table,
+ },
+#endif
+ { .ctl_name = 0 }
+};
+
+static ctl_table binfmt_misc_table[] = {
  { .ctl_name = 0 }
 };

--
1.4.4.1.g278f
```

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

Subject: [PATCH 53/59] sysctl: Remove support for CTL_ANY
Posted by ebiederm on Tue, 16 Jan 2007 16:39:58 GMT
View Forum Message <> Reply to Message

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

There are currently no users in the kernel for CTL_ANY and it
only has effect on the binary interface which is practically
unused.

So this complicates sysctl lookups for no good reason so just remove it.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/linux/sysctl.h | 1 -
 kernel/sysctl.c | 2 +-
 2 files changed, 1 insertions(+), 2 deletions(-)

diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 63e1bac..c99e4bb 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -53,7 +53,6 @@ struct __sysctl_args {

 /* For internal pattern-matching use only: */
 #ifdef __KERNEL__
-#define CTL_ANY -1 /* Matches any name */
 #define CTL_NONE 0
 #define CTL_UNNUMBERED CTL_NONE /* sysctl without a binary number */
 #endif
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index 8da6647..e655b11 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -1166,7 +1166,7 @@ repeat:
  for (; table->ctl_name || table->procname; table++) {
   if (!table->ctl_name)
    continue;
- if (n == table->ctl_name || table->ctl_name == CTL_ANY) {
+ if (n == table->ctl_name) {
    int error;
    if (table->child) {
     if (ctl_perm(table, 001))
--
1.4.4.1.g278f

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

Subject: [PATCH 54/59] sysctl: Remove support for directory strategy routines.
Posted by ebiederm on Tue, 16 Jan 2007 16:39:59 GMT
View Forum Message <> Reply to Message

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

parse_table has support for calling a strategy routine
when descending into a directory. To date no one has
used this functionality and the /proc/sys interface has
no analog to it.

So no one is using this functionality kill it and make
the binary sysctl code easier to follow.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 kernel/sysctl.c | 8 --------
 1 files changed, 0 insertions(+), 8 deletions(-)

diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index e655b11..2c3703d 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -1171,14 +1171,6 @@ repeat:
    if (table->child) {
     if (ctl_perm(table, 001))
      return -EPERM;
-    if (table->strategy) {
-     error = table->strategy(
-      table, name, nlen,
-      oldval, oldlenp,
-      newval, newlen);
-     if (error)
-      return error;
-    }
     name++;
     nlen--;
     table = table->child;
--
1.4.4.1.g278f

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

Subject: [PATCH 55/59] sysctl: Remove insert_at_head from register_sysctl

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

The semantic effect of insert_at_head is that it would allow
new registered sysctl entries to override existing sysctl entries
of the same name. Which is pain for caching and the proc interface
never implemented.

I have done an audit and discovered that none of the current
users of register_sysctl care as (excpet for directories) they
do not register duplicate sysctl entries.

So this patch simply removes the support for overriding
existing entries in the sys_sysctl interface since no one
uses it or cares and it makes future enhancments harder.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
arch/arm/kernel/isa.c | 2 +-
arch/frv/kernel/pm.c | 2 +-
arch/frv/kernel/sysctl.c | 2 +-
arch/ia64/kernel/crash.c | 2 +-
arch/ia64/kernel/perfmon.c | 2 +-
arch/ia64/sn/kernel/xpc_main.c | 2 +-
arch/mips/au1000/common/power.c | 2 +-
arch/mips/lasat/sysctl.c | 2 +-
arch/powerpc/kernel/idle.c | 2 +-
arch/ppc/kernel/ppc_htab.c | 2 +-
arch/s390/appldata/appldata_base.c | 4 ++--
arch/s390/kernel/debug.c | 2 +-
arch/s390/mm/cmm.c | 2 +-
arch/sh64/kernel/traps.c | 2 +-
arch/x86_64/ia32/ia32_binfmt.c | 2 +-
arch/x86_64/kernel/vsyscall.c | 2 +-
arch/x86_64/mm/init.c | 2 +-
drivers/cdrom/cdrom.c | 2 +-
drivers/char/hpet.c | 2 +-
drivers/char/ipmi/ipmi_poweroff.c | 2 +-
drivers/char/rtc.c | 2 +-
drivers/macintosh/mac_hid.c | 2 +-
drivers/md/md.c | 2 +-
drivers/net/wireless/arlan-proc.c | 2 +-
drivers/parport/procfs.c | 6 +++---
drivers/scsi/scsi_sysctl.c | 2 +-
fs/coda/sysctl.c | 2 +-
fs/dquot.c | 2 +-
fs/lockd/svc.c | 2 +-

```
 fs/nfs/sysctl.c                          |    2 +-
 fs/ntfs/sysctl.c                         |    2 +-
 fs/ocfs2/cluster/nodemanager.c           |    2 +-
 fs/xfs/linux-2.6/xfs_sysctl.c            |    2 +-
 include/linux/sysctl.h                   |    4 ++--
 ipc/ipc_sysctl.c                         |    2 +-
 ipc/mqueue.c                             |    2 +-
 kernel/sysctl.c                          |    9 ++-------
 kernel/utsname_sysctl.c                  |    2 +-
 net/appletalk/sysctl_net_atalk.c         |    2 +-
 net/ax25/sysctl_net_ax25.c               |    2 +-
 net/bridge/br_netfilter.c                |    2 +-
 net/core/neighbour.c                     |    2 +-
 net/dccp/sysctl.c                        |    2 +-
 net/decnet/dn_dev.c                      |    2 +-
 net/decnet/sysctl_net_decnet.c           |    2 +-
 net/ipv4/devinet.c                       |    4 ++--
 net/ipv4/ipvs/ip_vs_ctl.c                |    2 +-
 net/ipv4/ipvs/ip_vs_lblc.c               |    2 +-
 net/ipv4/ipvs/ip_vs_lblcr.c              |    2 +-
 net/ipv4/netfilter/ip_conntrack_proto_sctp.c |    2 +-
 net/ipv4/netfilter/ip_conntrack_standalone.c |    2 +-
 net/ipv4/netfilter/ip_queue.c            |    2 +-
 net/ipv6/addrconf.c                      |    4 ++--
 net/ipv6/netfilter/ip6_queue.c           |    2 +-
 net/ipv6/sysctl_net_ipv6.c               |    2 +-
 net/ipx/sysctl_net_ipx.c                 |    2 +-
 net/irda/irsysctl.c                      |    2 +-
 net/llc/sysctl_net_llc.c                 |    2 +-
 net/netfilter/nf_conntrack_standalone.c  |    2 +-
 net/netfilter/nf_sysctl.c                |    2 +-
 net/netrom/sysctl_net_netrom.c           |    2 +-
 net/rose/sysctl_net_rose.c               |    2 +-
 net/rxrpc/sysctl.c                       |    2 +-
 net/sctp/sysctl.c                        |    2 +-
 net/sunrpc/sysctl.c                      |    2 +-
 net/sunrpc/xprtsock.c                    |    2 +-
 net/unix/sysctl_net_unix.c               |    2 +-
 net/x25/sysctl_net_x25.c                 |    2 +-
 68 files changed, 75 insertions(+), 80 deletions(-)

diff --git a/arch/arm/kernel/isa.c b/arch/arm/kernel/isa.c
index 54bbd9f..50a30bc 100644
--- a/arch/arm/kernel/isa.c
+++ b/arch/arm/kernel/isa.c
@@ -70,5 +70,5 @@ register_isa_ports(unsigned int membase, unsigned int portbase, unsigned
int por
  isa_membase = membase;
```

```
  isa_portbase = portbase;
  isa_portshift = portshift;
- isa_sysctl_header = register_sysctl_table(ctl_bus, 0);
+ isa_sysctl_header = register_sysctl_table(ctl_bus);
 }
diff --git a/arch/frv/kernel/pm.c b/arch/frv/kernel/pm.c
index aa50333..c57ce3f 100644
--- a/arch/frv/kernel/pm.c
+++ b/arch/frv/kernel/pm.c
@@ -455,7 +455,7 @@ static struct ctl_table pm_dir_table[] =
  */
 static int __init pm_init(void)
 {
- register_sysctl_table(pm_dir_table, 0);
+ register_sysctl_table(pm_dir_table);
  return 0;
 }

diff --git a/arch/frv/kernel/sysctl.c b/arch/frv/kernel/sysctl.c
index 577ad16..3e9d7e0 100644
--- a/arch/frv/kernel/sysctl.c
+++ b/arch/frv/kernel/sysctl.c
@@ -216,7 +216,7 @@ static struct ctl_table frv_dir_table[] =
  */
 static int __init frv_sysctl_init(void)
 {
- register_sysctl_table(frv_dir_table, 0);
+ register_sysctl_table(frv_dir_table);
  return 0;
 }

diff --git a/arch/ia64/kernel/crash.c b/arch/ia64/kernel/crash.c
index bc2f64d..8a2cb76 100644
--- a/arch/ia64/kernel/crash.c
+++ b/arch/ia64/kernel/crash.c
@@ -214,7 +214,7 @@ machine_crash_setup(void)
  if((ret = register_die_notifier(&kdump_init_notifier_nb)) != 0)
   return ret;
 #ifdef CONFIG_SYSCTL
- register_sysctl_table(sys_table, 0);
+ register_sysctl_table(sys_table);
 #endif
  return 0;
 }
diff --git a/arch/ia64/kernel/perfmon.c b/arch/ia64/kernel/perfmon.c
index 8c679ab..8a15377 100644
--- a/arch/ia64/kernel/perfmon.c
+++ b/arch/ia64/kernel/perfmon.c
```

```
@@ -6727,7 +6727,7 @@ pfm_init(void)
  /*
   * create /proc/sys/kernel/perfmon (for debugging purposes)
   */
- pfm_sysctl_header = register_sysctl_table(pfm_sysctl_root, 0);
+ pfm_sysctl_header = register_sysctl_table(pfm_sysctl_root);

  /*
   * initialize all our spinlocks
diff --git a/arch/ia64/sn/kernel/xpc_main.c b/arch/ia64/sn/kernel/xpc_main.c
index e04f7b5..68355ef 100644
--- a/arch/ia64/sn/kernel/xpc_main.c
+++ b/arch/ia64/sn/kernel/xpc_main.c
@@ -1241,7 +1241,7 @@ xpc_init(void)
  snprintf(xpc_part->bus_id, BUS_ID_SIZE, "part");
  snprintf(xpc_chan->bus_id, BUS_ID_SIZE, "chan");

- xpc_sysctl = register_sysctl_table(xpc_sys_dir, 0);
+ xpc_sysctl = register_sysctl_table(xpc_sys_dir);

  /*
   * The first few fields of each entry of xpc_partitions[] need to
diff --git a/arch/mips/au1000/common/power.c b/arch/mips/au1000/common/power.c
index 31256b8..3901e8e 100644
--- a/arch/mips/au1000/common/power.c
+++ b/arch/mips/au1000/common/power.c
@@ -461,7 +461,7 @@ static struct ctl_table pm_dir_table[] = {
  */
 static int __init pm_init(void)
 {
- register_sysctl_table(pm_dir_table, 0);
+ register_sysctl_table(pm_dir_table);
  return 0;
 }

diff --git a/arch/mips/lasat/sysctl.c b/arch/mips/lasat/sysctl.c
index c04e82f..699ab18 100644
--- a/arch/mips/lasat/sysctl.c
+++ b/arch/mips/lasat/sysctl.c
@@ -432,7 +432,7 @@ static int __init lasat_register_sysctl(void)
  struct ctl_table_header *lasat_table_header;

  lasat_table_header =
-  register_sysctl_table(lasat_root_table, 0);
+  register_sysctl_table(lasat_root_table);

  return 0;
 }
```

```
diff --git a/arch/powerpc/kernel/idle.c b/arch/powerpc/kernel/idle.c
index 8b27bb1..6e7f509 100644
--- a/arch/powerpc/kernel/idle.c
+++ b/arch/powerpc/kernel/idle.c
@@ -125,7 +125,7 @@ static ctl_table powersave_nap_sysctl_root[] = {
 static int __init
 register_powersave_nap_sysctl(void)
 {
- register_sysctl_table(powersave_nap_sysctl_root, 0);
+ register_sysctl_table(powersave_nap_sysctl_root);

  return 0;
 }
diff --git a/arch/ppc/kernel/ppc_htab.c b/arch/ppc/kernel/ppc_htab.c
index 77b20ff..0a7e42d 100644
--- a/arch/ppc/kernel/ppc_htab.c
+++ b/arch/ppc/kernel/ppc_htab.c
@@ -457,7 +457,7 @@ static ctl_table htab_sysctl_root[] = {
 static int __init
 register_ppc_htab_sysctl(void)
 {
- register_sysctl_table(htab_sysctl_root, 0);
+ register_sysctl_table(htab_sysctl_root);

  return 0;
 }
diff --git a/arch/s390/appldata/appldata_base.c b/arch/s390/appldata/appldata_base.c
index cdc4109..b1ff93a 100644
--- a/arch/s390/appldata/appldata_base.c
+++ b/arch/s390/appldata/appldata_base.c
@@ -506,7 +506,7 @@ int appldata_register_ops(struct appldata_ops *ops)

  ops->ctl_table[3].ctl_name = 0;

- ops->sysctl_header = register_sysctl_table(ops->ctl_table,0);
+ ops->sysctl_header = register_sysctl_table(ops->ctl_table);

  P_INFO("%s-ops registered!\n", ops->name);
  return 0;
@@ -606,7 +606,7 @@ static int __init appldata_init(void)
  /* Register cpu hotplug notifier */
  register_hotcpu_notifier(&appldata_nb);

- appldata_sysctl_header = register_sysctl_table(appldata_dir_table, 0);
+ appldata_sysctl_header = register_sysctl_table(appldata_dir_table);
 #ifdef MODULE
  appldata_dir_table[0].de->owner = THIS_MODULE;
  appldata_table[0].de->owner = THIS_MODULE;
```

```
diff --git a/arch/s390/kernel/debug.c b/arch/s390/kernel/debug.c
index d38cb27..00f0382 100644
--- a/arch/s390/kernel/debug.c
+++ b/arch/s390/kernel/debug.c
@@ -1053,7 +1053,7 @@ __init debug_init(void)
 {
  int rc = 0;

- s390dbf_sysctl_header = register_sysctl_table(s390dbf_dir_table, 0);
+ s390dbf_sysctl_header = register_sysctl_table(s390dbf_dir_table);
  down(&debug_lock);
  debug_debugfs_root_entry = debugfs_create_dir(DEBUG_DIR_ROOT,NULL);
  printk(KERN_INFO "debug: Initialization complete\n");
diff --git a/arch/s390/mm/cmm.c b/arch/s390/mm/cmm.c
index 5f83a3f..8a5c71f 100644
--- a/arch/s390/mm/cmm.c
+++ b/arch/s390/mm/cmm.c
@@ -418,7 +418,7 @@ cmm_init (void)
  int rc = -ENOMEM;

 #ifdef CONFIG_CMM_PROC
- cmm_sysctl_header = register_sysctl_table(cmm_dir_table, 0);
+ cmm_sysctl_header = register_sysctl_table(cmm_dir_table);
  if (!cmm_sysctl_header)
   goto out;
 #endif
diff --git a/arch/sh64/kernel/traps.c b/arch/sh64/kernel/traps.c
index 02cca74..c346d7e 100644
--- a/arch/sh64/kernel/traps.c
+++ b/arch/sh64/kernel/traps.c
@@ -960,7 +960,7 @@ static ctl_table sh64_root[] = {
 static struct ctl_table_header *sysctl_header;
 static int __init init_sysctl(void)
 {
- sysctl_header = register_sysctl_table(sh64_root, 0);
+ sysctl_header = register_sysctl_table(sh64_root);
  return 0;
 }

diff --git a/arch/x86_64/ia32/ia32_binfmt.c b/arch/x86_64/ia32/ia32_binfmt.c
index 644b203..aab2c91 100644
--- a/arch/x86_64/ia32/ia32_binfmt.c
+++ b/arch/x86_64/ia32/ia32_binfmt.c
@@ -418,7 +418,7 @@ static ctl_table abi_root_table2[] = {

 static __init int ia32_binfmt_init(void)
 {
- register_sysctl_table(abi_root_table2, 0);
```

```
+ register_sysctl_table(abi_root_table2);
  return 0;
 }
 __initcall(ia32_binfmt_init);
diff --git a/arch/x86_64/kernel/vsyscall.c b/arch/x86_64/kernel/vsyscall.c
index c0e2b48..313dc6a 100644
--- a/arch/x86_64/kernel/vsyscall.c
+++ b/arch/x86_64/kernel/vsyscall.c
@@ -301,7 +301,7 @@ static int __init vsyscall_init(void)
  BUG_ON((unsigned long) &vgetcpu != VSYSCALL_ADDR(__NR_vgetcpu));
  map_vsyscall();
 #ifdef CONFIG_SYSCTL
- register_sysctl_table(kernel_root_table2, 0);
+ register_sysctl_table(kernel_root_table2);
 #endif
  on_each_cpu(cpu_vsyscall_init, NULL, 0, 1);
  hotcpu_notifier(cpu_vsyscall_notifier, 0);
diff --git a/arch/x86_64/mm/init.c b/arch/x86_64/mm/init.c
index a04535d..079bcaa 100644
--- a/arch/x86_64/mm/init.c
+++ b/arch/x86_64/mm/init.c
@@ -734,7 +734,7 @@ static ctl_table debug_root_table2[] = {

 static __init int x8664_sysctl_init(void)
 {
- register_sysctl_table(debug_root_table2, 0);
+ register_sysctl_table(debug_root_table2);
  return 0;
 }
 __initcall(x8664_sysctl_init);
diff --git a/drivers/cdrom/cdrom.c b/drivers/cdrom/cdrom.c
index 14f72c4..b36f44d 100644
--- a/drivers/cdrom/cdrom.c
+++ b/drivers/cdrom/cdrom.c
@@ -3553,7 +3553,7 @@ static void cdrom_sysctl_register(void)
  if (initialized == 1)
   return;

- cdrom_sysctl_header = register_sysctl_table(cdrom_root_table, 0);
+ cdrom_sysctl_header = register_sysctl_table(cdrom_root_table);

  /* set the defaults */
  cdrom_sysctl_settings.autoclose = autoclose;
diff --git a/drivers/char/hpet.c b/drivers/char/hpet.c
index 81be1db..0be700f 100644
--- a/drivers/char/hpet.c
+++ b/drivers/char/hpet.c
@@ -1018,7 +1018,7 @@ static int __init hpet_init(void)
```

```
  if (result < 0)
   return -ENODEV;

- sysctl_header = register_sysctl_table(dev_root, 0);
+ sysctl_header = register_sysctl_table(dev_root);

  result = acpi_bus_register_driver(&hpet_acpi_driver);
  if (result < 0) {
diff --git a/drivers/char/ipmi/ipmi_poweroff.c b/drivers/char/ipmi/ipmi_poweroff.c
index b3ae65e..e02893b 100644
--- a/drivers/char/ipmi/ipmi_poweroff.c
+++ b/drivers/char/ipmi/ipmi_poweroff.c
@@ -686,7 +686,7 @@ static int ipmi_poweroff_init (void)
   printk(KERN_INFO PFX "Power cycle is enabled.\n");

 #ifdef CONFIG_PROC_FS
- ipmi_table_header = register_sysctl_table(ipmi_root_table, 0);
+ ipmi_table_header = register_sysctl_table(ipmi_root_table);
  if (!ipmi_table_header) {
   printk(KERN_ERR PFX "Unable to register powercycle sysctl\n");
   rv = -ENOMEM;
diff --git a/drivers/char/rtc.c b/drivers/char/rtc.c
index df11289..98a2f5e 100644
--- a/drivers/char/rtc.c
+++ b/drivers/char/rtc.c
@@ -316,7 +316,7 @@ static struct ctl_table_header *sysctl_header;

 static int __init init_sysctl(void)
 {
-    sysctl_header = register_sysctl_table(dev_root, 0);
+    sysctl_header = register_sysctl_table(dev_root);
     return 0;
 }

diff --git a/drivers/macintosh/mac_hid.c b/drivers/macintosh/mac_hid.c
index c676740..030399f 100644
--- a/drivers/macintosh/mac_hid.c
+++ b/drivers/macintosh/mac_hid.c
@@ -138,7 +138,7 @@ int __init mac_hid_init(void)
   return err;

 #if defined(CONFIG_SYSCTL)
- mac_hid_sysctl_header = register_sysctl_table(mac_hid_root_dir, 0);
+ mac_hid_sysctl_header = register_sysctl_table(mac_hid_root_dir);
 #endif /* CONFIG_SYSCTL */

  return 0;
diff --git a/drivers/md/md.c b/drivers/md/md.c
```

```
index 966e8be..72b378e 100644
--- a/drivers/md/md.c
+++ b/drivers/md/md.c
@@ -5551,7 +5551,7 @@ static int __init md_init(void)
        md_probe, NULL, NULL);

  register_reboot_notifier(&md_notifier);
- raid_table_header = register_sysctl_table(raid_root_table, 0);
+ raid_table_header = register_sysctl_table(raid_root_table);

  md_geninit();
  return (0);
diff --git a/drivers/net/wireless/arlan-proc.c b/drivers/net/wireless/arlan-proc.c
index 20499a6..015abd9 100644
--- a/drivers/net/wireless/arlan-proc.c
+++ b/drivers/net/wireless/arlan-proc.c
@@ -1244,7 +1244,7 @@ int __init init_arlan_proc(void)
   return 0;
  for (i = 0; i < MAX_ARLANS && arlan_device[i]; i++)
   arlan_table[i].ctl_name = i + 1;
- arlan_device_sysctl_header = register_sysctl_table(arlan_root_table, 0);
+ arlan_device_sysctl_header = register_sysctl_table(arlan_root_table);
  if (!arlan_device_sysctl_header)
   return -1;

diff --git a/drivers/parport/procfs.c b/drivers/parport/procfs.c
index 5337789..afdd392 100644
--- a/drivers/parport/procfs.c
+++ b/drivers/parport/procfs.c
@@ -518,7 +518,7 @@ int parport_proc_register(struct parport *port)
  t->parport_dir[0].child = t->port_dir;
  t->dev_dir[0].child = t->parport_dir;

- t->sysctl_header = register_sysctl_table(t->dev_dir, 0);
+ t->sysctl_header = register_sysctl_table(t->dev_dir);
  if (t->sysctl_header == NULL) {
   kfree(t);
   t = NULL;
@@ -574,7 +574,7 @@ int parport_device_proc_register(struct pardevice *device)
  t->device_dir[0].child = t->vars;
  t->vars[0].data = &device->timeslice;

- t->sysctl_header = register_sysctl_table(t->dev_dir, 0);
+ t->sysctl_header = register_sysctl_table(t->dev_dir);
  if (t->sysctl_header == NULL) {
   kfree(t);
   t = NULL;
@@ -597,7 +597,7 @@ int parport_device_proc_unregister(struct pardevice *device)
```

```
 static int __init parport_default_proc_register(void)
 {
  parport_default_sysctl_table.sysctl_header =
-  register_sysctl_table(parport_default_sysctl_table.dev_dir, 0);
+  register_sysctl_table(parport_default_sysctl_table.dev_dir);
  return 0;
 }

diff --git a/drivers/scsi/scsi_sysctl.c b/drivers/scsi/scsi_sysctl.c
index b16b775..6cfaaa2 100644
--- a/drivers/scsi/scsi_sysctl.c
+++ b/drivers/scsi/scsi_sysctl.c
@@ -41,7 +41,7 @@ static struct ctl_table_header *scsi_table_header;

 int __init scsi_init_sysctl(void)
 {
- scsi_table_header = register_sysctl_table(scsi_root_table, 0);
+ scsi_table_header = register_sysctl_table(scsi_root_table);
  if (!scsi_table_header)
   return -ENOMEM;
  return 0;
diff --git a/fs/coda/sysctl.c b/fs/coda/sysctl.c
index df682e2..77d7563 100644
--- a/fs/coda/sysctl.c
+++ b/fs/coda/sysctl.c
@@ -269,7 +269,7 @@ void coda_sysctl_init(void)

 #ifdef CONFIG_SYSCTL
  if ( !fs_table_header )
-  fs_table_header = register_sysctl_table(fs_table, 0);
+  fs_table_header = register_sysctl_table(fs_table);
 #endif
 }

diff --git a/fs/dquot.c b/fs/dquot.c
index 0952cc4..9a25927 100644
--- a/fs/dquot.c
+++ b/fs/dquot.c
@@ -1822,7 +1822,7 @@ static int __init dquot_init(void)

  printk(KERN_NOTICE "VFS: Disk quotas %s\n", __DQUOT_VERSION__);

- register_sysctl_table(sys_table, 0);
+ register_sysctl_table(sys_table);

  dquot_cachep = kmem_cache_create("dquot",
    sizeof(struct dquot), sizeof(unsigned long) * 4,
diff --git a/fs/lockd/svc.c b/fs/lockd/svc.c
```

```
index 8ca1808..67aa93e 100644
--- a/fs/lockd/svc.c
+++ b/fs/lockd/svc.c
@@ -506,7 +506,7 @@ module_param(nsm_use_hostnames, bool, 0644);

 static int __init init_nlm(void)
 {
- nlm_sysctl_table = register_sysctl_table(nlm_sysctl_root, 0);
+ nlm_sysctl_table = register_sysctl_table(nlm_sysctl_root);
  return nlm_sysctl_table ? 0 : -ENOMEM;
 }

diff --git a/fs/nfs/sysctl.c b/fs/nfs/sysctl.c
index 3ea50ac..fcdcafb 100644
--- a/fs/nfs/sysctl.c
+++ b/fs/nfs/sysctl.c
@@ -75,7 +75,7 @@ static ctl_table nfs_cb_sysctl_root[] = {

 int nfs_register_sysctl(void)
 {
- nfs_callback_sysctl_table = register_sysctl_table(nfs_cb_sysctl_root, 0);
+ nfs_callback_sysctl_table = register_sysctl_table(nfs_cb_sysctl_root);
  if (nfs_callback_sysctl_table == NULL)
   return -ENOMEM;
  return 0;
diff --git a/fs/ntfs/sysctl.c b/fs/ntfs/sysctl.c
index bc217de..7edb370 100644
--- a/fs/ntfs/sysctl.c
+++ b/fs/ntfs/sysctl.c
@@ -70,7 +70,7 @@ int ntfs_sysctl(int add)
 {
  if (add) {
   BUG_ON(sysctls_root_table);
-  sysctls_root_table = register_sysctl_table(sysctls_root, 0);
+  sysctls_root_table = register_sysctl_table(sysctls_root);
   if (!sysctls_root_table)
    return -ENOMEM;
 #ifdef CONFIG_PROC_FS
diff --git a/fs/ocfs2/cluster/nodemanager.c b/fs/ocfs2/cluster/nodemanager.c
index df763c7..9f5ad0f 100644
--- a/fs/ocfs2/cluster/nodemanager.c
+++ b/fs/ocfs2/cluster/nodemanager.c
@@ -922,7 +922,7 @@ static int __init init_o2nm(void)
  o2hb_init();
  o2net_init();

- ocfs2_table_header = register_sysctl_table(ocfs2_root_table, 0);
+ ocfs2_table_header = register_sysctl_table(ocfs2_root_table);
```

```
  if (!ocfs2_table_header) {
   printk(KERN_ERR "nodemanager: unable to register sysctl\n");
   ret = -ENOMEM; /* or something. */
diff --git a/fs/xfs/linux-2.6/xfs_sysctl.c b/fs/xfs/linux-2.6/xfs_sysctl.c
index 5a0eefc..3ac4dab 100644
--- a/fs/xfs/linux-2.6/xfs_sysctl.c
+++ b/fs/xfs/linux-2.6/xfs_sysctl.c
@@ -251,7 +251,7 @@ STATIC ctl_table xfs_root_table[] = {
 void
 xfs_sysctl_register(void)
 {
- xfs_table_header = register_sysctl_table(xfs_root_table, 0);
+ xfs_table_header = register_sysctl_table(xfs_root_table);
 }

 void
diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index c99e4bb..6113f3b 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -1034,8 +1034,8 @@ struct ctl_table_header
  struct completion *unregistering;
 };

-struct ctl_table_header * register_sysctl_table(ctl_table * table,
-       int insert_at_head);
+struct ctl_table_header * register_sysctl_table(ctl_table * table);
+
 void unregister_sysctl_table(struct ctl_table_header * table);

 #else /* __KERNEL__ */
diff --git a/ipc/ipc_sysctl.c b/ipc/ipc_sysctl.c
index 9018009..a491c60 100644
--- a/ipc/ipc_sysctl.c
+++ b/ipc/ipc_sysctl.c
@@ -175,7 +175,7 @@ static struct ctl_table ipc_root_table[] = {

 static int __init ipc_sysctl_init(void)
 {
- register_sysctl_table(ipc_root_table, 0);
+ register_sysctl_table(ipc_root_table);
  return 0;
 }

diff --git a/ipc/mqueue.c b/ipc/mqueue.c
index 02717f7..00b842a 100644
--- a/ipc/mqueue.c
+++ b/ipc/mqueue.c
```

```
@@ -1255,7 +1255,7 @@ static int __init init_mqueue_fs(void)
   return -ENOMEM;

  /* ignore failues - they are not fatal */
- mq_sysctl_table = register_sysctl_table(mq_sysctl_root, 0);
+ mq_sysctl_table = register_sysctl_table(mq_sysctl_root);

  error = register_filesystem(&mqueue_fs_type);
  if (error)
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index 2c3703d..5beee1f 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -1240,7 +1240,6 @@ int do_sysctl_strategy (ctl_table *table,
 /**
  * register_sysctl_table - register a sysctl hierarchy
  * @table: the top-level table structure
- * @insert_at_head: whether the entry should be inserted in front or at the end
  *
  * Register a sysctl table hierarchy. @table should be a filled in ctl_table
  * array. An entry with a ctl_name of 0 terminates the table.
@@ -1306,8 +1305,7 @@ int do_sysctl_strategy (ctl_table *table,
  * This routine returns %NULL on a failure to register, and a pointer
  * to the table header on success.
  */
-struct ctl_table_header *register_sysctl_table(ctl_table * table,
-                                               int insert_at_head)
+struct ctl_table_header *register_sysctl_table(ctl_table * table)
 {
  struct ctl_table_header *tmp;
  tmp = kmalloc(sizeof(struct ctl_table_header), GFP_KERNEL);
@@ -1318,10 +1316,7 @@ struct ctl_table_header *register_sysctl_table(ctl_table * table,
  tmp->used = 0;
  tmp->unregistering = NULL;
  spin_lock(&sysctl_lock);
- if (insert_at_head)
-  list_add(&tmp->ctl_entry, &root_table_header.ctl_entry);
- else
-  list_add_tail(&tmp->ctl_entry, &root_table_header.ctl_entry);
+ list_add_tail(&tmp->ctl_entry, &root_table_header.ctl_entry);
  spin_unlock(&sysctl_lock);
 #ifdef CONFIG_PROC_SYSCTL
  register_proc_table(table, proc_sys_root, tmp);
diff --git a/kernel/utsname_sysctl.c b/kernel/utsname_sysctl.c
index 324aa13..f22b9db 100644
--- a/kernel/utsname_sysctl.c
+++ b/kernel/utsname_sysctl.c
@@ -139,7 +139,7 @@ static struct ctl_table uts_root_table[] = {
```

```
 static int __init utsname_sysctl_init(void)
 {
- register_sysctl_table(uts_root_table, 0);
+ register_sysctl_table(uts_root_table);
  return 0;
 }

diff --git a/net/appletalk/sysctl_net_atalk.c b/net/appletalk/sysctl_net_atalk.c
index 4f806b6..7df1778 100644
--- a/net/appletalk/sysctl_net_atalk.c
+++ b/net/appletalk/sysctl_net_atalk.c
@@ -73,7 +73,7 @@ static struct ctl_table_header *atalk_table_header;

 void atalk_register_sysctl(void)
 {
- atalk_table_header = register_sysctl_table(atalk_root_table, 0);
+ atalk_table_header = register_sysctl_table(atalk_root_table);
 }

 void atalk_unregister_sysctl(void)
diff --git a/net/ax25/sysctl_net_ax25.c b/net/ax25/sysctl_net_ax25.c
index afdba04..443a836 100644
--- a/net/ax25/sysctl_net_ax25.c
+++ b/net/ax25/sysctl_net_ax25.c
@@ -245,7 +245,7 @@ void ax25_register_sysctl(void)

  ax25_dir_table[0].child = ax25_table;

- ax25_table_header = register_sysctl_table(ax25_root_table, 0);
+ ax25_table_header = register_sysctl_table(ax25_root_table);
 }

 void ax25_unregister_sysctl(void)
diff --git a/net/bridge/br_netfilter.c b/net/bridge/br_netfilter.c
index ea3337a..77998b6 100644
--- a/net/bridge/br_netfilter.c
+++ b/net/bridge/br_netfilter.c
@@ -966,7 +966,7 @@ int br_netfilter_init(void)
  }

 #ifdef CONFIG_SYSCTL
- brnf_sysctl_header = register_sysctl_table(brnf_net_table, 0);
+ brnf_sysctl_header = register_sysctl_table(brnf_net_table);
  if (brnf_sysctl_header == NULL) {
   printk(KERN_WARNING
          "br_netfilter: can't register to sysctl.\n");
diff --git a/net/core/neighbour.c b/net/core/neighbour.c
```

```
index e7300b6..8437678 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -2701,7 +2701,7 @@ int neigh_sysctl_register(struct net_device *dev, struct neigh_parms
*p,
  t->neigh_proto_dir[0].child    = t->neigh_neigh_dir;
  t->neigh_root_dir[0].child     = t->neigh_proto_dir;

- t->sysctl_header = register_sysctl_table(t->neigh_root_dir, 0);
+ t->sysctl_header = register_sysctl_table(t->neigh_root_dir);
  if (!t->sysctl_header) {
   err = -ENOBUFS;
   goto free_procname;
diff --git a/net/dccp/sysctl.c b/net/dccp/sysctl.c
index 3391631..1260aab 100644
--- a/net/dccp/sysctl.c
+++ b/net/dccp/sysctl.c
@@ -127,7 +127,7 @@ static struct ctl_table_header *dccp_table_header;

 int __init dccp_sysctl_init(void)
 {
- dccp_table_header = register_sysctl_table(dccp_root_table, 0);
+ dccp_table_header = register_sysctl_table(dccp_root_table);

  return dccp_table_header != NULL ? 0 : -ENOMEM;
 }
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index fc6f3c0..baaa02e 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -282,7 +282,7 @@ static void dn_dev_sysctl_register(struct net_device *dev, struct
dn_dev_parms *
  t->dn_dev_root_dir[0].de = NULL;
  t->dn_dev_vars[0].extra1 = (void *)dev;

- t->sysctl_header = register_sysctl_table(t->dn_dev_root_dir, 0);
+ t->sysctl_header = register_sysctl_table(t->dn_dev_root_dir);
  if (t->sysctl_header == NULL)
   kfree(t);
  else
diff --git a/net/decnet/sysctl_net_decnet.c b/net/decnet/sysctl_net_decnet.c
index 81469fd..37fff9a 100644
--- a/net/decnet/sysctl_net_decnet.c
+++ b/net/decnet/sysctl_net_decnet.c
@@ -491,7 +491,7 @@ static ctl_table dn_root_table[] = {

 void dn_register_sysctl(void)
 {
```

```
- dn_table_header = register_sysctl_table(dn_root_table, 0);
+ dn_table_header = register_sysctl_table(dn_root_table);
 }

 void dn_unregister_sysctl(void)
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 480ace9..b731a0c 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1603,7 +1603,7 @@ static void devinet_sysctl_register(struct in_device *in_dev,
  t->devinet_root_dir[0].child  = t->devinet_proto_dir;
  t->devinet_root_dir[0].de     = NULL;

- t->sysctl_header = register_sysctl_table(t->devinet_root_dir, 0);
+ t->sysctl_header = register_sysctl_table(t->devinet_root_dir);
  if (!t->sysctl_header)
      goto free_procname;

@@ -1637,7 +1637,7 @@ void __init devinet_init(void)
  rtnetlink_links[PF_INET] = inet_rtnetlink_table;
 #ifdef CONFIG_SYSCTL
  devinet_sysctl.sysctl_header =
-  register_sysctl_table(devinet_sysctl.devinet_root_dir, 0);
+  register_sysctl_table(devinet_sysctl.devinet_root_dir);
  devinet_sysctl_register(NULL, &ipv4_devconf_dflt);
 #endif
 }
diff --git a/net/ipv4/ipvs/ip_vs_ctl.c b/net/ipv4/ipvs/ip_vs_ctl.c
index 9b93338..c4e4237 100644
--- a/net/ipv4/ipvs/ip_vs_ctl.c
+++ b/net/ipv4/ipvs/ip_vs_ctl.c
@@ -2359,7 +2359,7 @@ int ip_vs_control_init(void)
  proc_net_fops_create("ip_vs", 0, &ip_vs_info_fops);
  proc_net_fops_create("ip_vs_stats",0, &ip_vs_stats_fops);

- sysctl_header = register_sysctl_table(vs_root_table, 0);
+ sysctl_header = register_sysctl_table(vs_root_table);

  /* Initialize ip_vs_svc_table, ip_vs_svc_fwm_table, ip_vs_rtable */
  for(idx = 0; idx < IP_VS_SVC_TAB_SIZE; idx++)  {
diff --git a/net/ipv4/ipvs/ip_vs_lblc.c b/net/ipv4/ipvs/ip_vs_lblc.c
index a4385a2..0e9cdfc 100644
--- a/net/ipv4/ipvs/ip_vs_lblc.c
+++ b/net/ipv4/ipvs/ip_vs_lblc.c
@@ -583,7 +583,7 @@ static struct ip_vs_scheduler ip_vs_lblc_scheduler =
 static int __init ip_vs_lblc_init(void)
 {
  INIT_LIST_HEAD(&ip_vs_lblc_scheduler.n_list);
```

```
- sysctl_header = register_sysctl_table(lblc_root_table, 0);
+ sysctl_header = register_sysctl_table(lblc_root_table);
  return register_ip_vs_scheduler(&ip_vs_lblc_scheduler);
 }

diff --git a/net/ipv4/ipvs/ip_vs_lblcr.c b/net/ipv4/ipvs/ip_vs_lblcr.c
index fe1af5d..22004f8 100644
--- a/net/ipv4/ipvs/ip_vs_lblcr.c
+++ b/net/ipv4/ipvs/ip_vs_lblcr.c
@@ -841,7 +841,7 @@ static struct ip_vs_scheduler ip_vs_lblcr_scheduler =
 static int __init ip_vs_lblcr_init(void)
 {
  INIT_LIST_HEAD(&ip_vs_lblcr_scheduler.n_list);
- sysctl_header = register_sysctl_table(lblcr_root_table, 0);
+ sysctl_header = register_sysctl_table(lblcr_root_table);
 #ifdef CONFIG_IP_VS_LBLCR_DEBUG
  proc_net_create("ip_vs_lblcr", 0, ip_vs_lblcr_getinfo);
 #endif
diff --git a/net/ipv4/netfilter/ip_conntrack_proto_sctp.c
b/net/ipv4/netfilter/ip_conntrack_proto_sctp.c
index 2443322..44513b4 100644
--- a/net/ipv4/netfilter/ip_conntrack_proto_sctp.c
+++ b/net/ipv4/netfilter/ip_conntrack_proto_sctp.c
@@ -623,7 +623,7 @@ static int __init ip_conntrack_proto_sctp_init(void)
  }

 #ifdef CONFIG_SYSCTL
- ip_ct_sysctl_header = register_sysctl_table(ip_ct_net_table, 0);
+ ip_ct_sysctl_header = register_sysctl_table(ip_ct_net_table);
  if (ip_ct_sysctl_header == NULL) {
   ret = -ENOMEM;
   printk("ip_conntrack_proto_sctp: can't register to sysctl.\n");
diff --git a/net/ipv4/netfilter/ip_conntrack_standalone.c
b/net/ipv4/netfilter/ip_conntrack_standalone.c
index 86efb54..9d89469 100644
--- a/net/ipv4/netfilter/ip_conntrack_standalone.c
+++ b/net/ipv4/netfilter/ip_conntrack_standalone.c
@@ -849,7 +849,7 @@ static int __init ip_conntrack_standalone_init(void)
   goto cleanup_proc_stat;
  }
 #ifdef CONFIG_SYSCTL
- ip_ct_sysctl_header = register_sysctl_table(ip_ct_net_table, 0);
+ ip_ct_sysctl_header = register_sysctl_table(ip_ct_net_table);
  if (ip_ct_sysctl_header == NULL) {
   printk("ip_conntrack: can't register to sysctl.\n");
   ret = -ENOMEM;
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index cd520df..3446d4a 100644
```

```
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -693,7 +693,7 @@ static int __init ip_queue_init(void)
  }

  register_netdevice_notifier(&ipq_dev_notifier);
- ipq_sysctl_header = register_sysctl_table(ipq_root_table, 0);
+ ipq_sysctl_header = register_sysctl_table(ipq_root_table);

  status = nf_register_queue_handler(PF_INET, &nfqh);
  if (status < 0) {
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 171e5b5..791aaba 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -4004,7 +4004,7 @@ static void addrconf_sysctl_register(struct inet6_dev *idev, struct
ipv6_devconf
  t->addrconf_root_dir[0].child = t->addrconf_proto_dir;
  t->addrconf_root_dir[0].de = NULL;

- t->sysctl_header = register_sysctl_table(t->addrconf_root_dir, 0);
+ t->sysctl_header = register_sysctl_table(t->addrconf_root_dir);
  if (t->sysctl_header == NULL)
   goto free_procname;
  else
@@ -4089,7 +4089,7 @@ int __init addrconf_init(void)
  rtnetlink_links[PF_INET6] = inet6_rtnetlink_table;
 #ifdef CONFIG_SYSCTL
  addrconf_sysctl.sysctl_header =
-  register_sysctl_table(addrconf_sysctl.addrconf_root_dir, 0);
+  register_sysctl_table(addrconf_sysctl.addrconf_root_dir);
  addrconf_sysctl_register(NULL, &ipv6_devconf_dflt);
 #endif

diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index d4d9f18..e774be7 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -683,7 +683,7 @@ static int __init ip6_queue_init(void)
  }

  register_netdevice_notifier(&ipq_dev_notifier);
- ipq_sysctl_header = register_sysctl_table(ipq_root_table, 0);
+ ipq_sysctl_header = register_sysctl_table(ipq_root_table);

  status = nf_register_queue_handler(PF_INET6, &nfqh);
  if (status < 0) {
diff --git a/net/ipv6/sysctl_net_ipv6.c b/net/ipv6/sysctl_net_ipv6.c
```

```
index 7a4639d..31e2494 100644
--- a/net/ipv6/sysctl_net_ipv6.c
+++ b/net/ipv6/sysctl_net_ipv6.c
@@ -107,7 +107,7 @@ static ctl_table ipv6_root_table[] = {

 void ipv6_sysctl_register(void)
 {
- ipv6_sysctl_header = register_sysctl_table(ipv6_root_table, 0);
+ ipv6_sysctl_header = register_sysctl_table(ipv6_root_table);
 }

 void ipv6_sysctl_unregister(void)
diff --git a/net/ipx/sysctl_net_ipx.c b/net/ipx/sysctl_net_ipx.c
index 0442f44..55a7d96 100644
--- a/net/ipx/sysctl_net_ipx.c
+++ b/net/ipx/sysctl_net_ipx.c
@@ -52,7 +52,7 @@ static struct ctl_table_header *ipx_table_header;

 void ipx_register_sysctl(void)
 {
- ipx_table_header = register_sysctl_table(ipx_root_table, 0);
+ ipx_table_header = register_sysctl_table(ipx_root_table);
 }

 void ipx_unregister_sysctl(void)
diff --git a/net/irda/irsysctl.c b/net/irda/irsysctl.c
index 86805c3..b60b72f 100644
--- a/net/irda/irsysctl.c
+++ b/net/irda/irsysctl.c
@@ -274,7 +274,7 @@ static struct ctl_table_header *irda_table_header;
  */
 int __init irda_sysctl_register(void)
 {
- irda_table_header = register_sysctl_table(irda_root_table, 0);
+ irda_table_header = register_sysctl_table(irda_root_table);
  if (!irda_table_header)
   return -ENOMEM;

diff --git a/net/llc/sysctl_net_llc.c b/net/llc/sysctl_net_llc.c
index 4aab676..154a3e6 100644
--- a/net/llc/sysctl_net_llc.c
+++ b/net/llc/sysctl_net_llc.c
@@ -116,7 +116,7 @@ static struct ctl_table_header *llc_table_header;

 int __init llc_sysctl_init(void)
 {
- llc_table_header = register_sysctl_table(llc_root_table, 0);
+ llc_table_header = register_sysctl_table(llc_root_table);
```

```
  return llc_table_header ? 0 : -ENOMEM;
 }
diff --git a/net/netfilter/nf_conntrack_standalone.c b/net/netfilter/nf_conntrack_standalone.c
index f1cb60f..2587b49 100644
--- a/net/netfilter/nf_conntrack_standalone.c
+++ b/net/netfilter/nf_conntrack_standalone.c
@@ -445,7 +445,7 @@ static int __init nf_conntrack_standalone_init(void)
  proc_stat->owner = THIS_MODULE;
 #endif
 #ifdef CONFIG_SYSCTL
- nf_ct_sysctl_header = register_sysctl_table(nf_ct_net_table, 0);
+ nf_ct_sysctl_header = register_sysctl_table(nf_ct_net_table);
  if (nf_ct_sysctl_header == NULL) {
   printk("nf_conntrack: can't register to sysctl.\n");
   ret = -ENOMEM;
diff --git a/net/netfilter/nf_sysctl.c b/net/netfilter/nf_sysctl.c
index 06ddddb..ee34589 100644
--- a/net/netfilter/nf_sysctl.c
+++ b/net/netfilter/nf_sysctl.c
@@ -56,7 +56,7 @@ nf_register_sysctl_table(struct ctl_table *path, struct ctl_table *table)
  path = path_dup(path, table);
  if (path == NULL)
   return NULL;
- header = register_sysctl_table(path, 0);
+ header = register_sysctl_table(path);
  if (header == NULL)
   path_free(path, table);
  return header;
diff --git a/net/netrom/sysctl_net_netrom.c b/net/netrom/sysctl_net_netrom.c
index 09f4246..4cbc309 100644
--- a/net/netrom/sysctl_net_netrom.c
+++ b/net/netrom/sysctl_net_netrom.c
@@ -192,7 +192,7 @@ static ctl_table nr_root_table[] = {

 void __init nr_register_sysctl(void)
 {
- nr_table_header = register_sysctl_table(nr_root_table, 0);
+ nr_table_header = register_sysctl_table(nr_root_table);
 }

 void nr_unregister_sysctl(void)
diff --git a/net/rose/sysctl_net_rose.c b/net/rose/sysctl_net_rose.c
index 0190a07..c30e1c7 100644
--- a/net/rose/sysctl_net_rose.c
+++ b/net/rose/sysctl_net_rose.c
@@ -160,7 +160,7 @@ static ctl_table rose_root_table[] = {
```

```
 void __init rose_register_sysctl(void)
 {
- rose_table_header = register_sysctl_table(rose_root_table, 0);
+ rose_table_header = register_sysctl_table(rose_root_table);
 }

 void rose_unregister_sysctl(void)
diff --git a/net/rxrpc/sysctl.c b/net/rxrpc/sysctl.c
index 6374df7..3bad91a 100644
--- a/net/rxrpc/sysctl.c
+++ b/net/rxrpc/sysctl.c
@@ -97,7 +97,7 @@ static ctl_table rxrpc_dir_sysctl_table[] = {
 int rxrpc_sysctl_init(void)
 {
 #ifdef CONFIG_SYSCTL
- rxrpc_sysctl = register_sysctl_table(rxrpc_dir_sysctl_table, 0);
+ rxrpc_sysctl = register_sysctl_table(rxrpc_dir_sysctl_table);
  if (!rxrpc_sysctl)
   return -ENOMEM;
 #endif /* CONFIG_SYSCTL */
diff --git a/net/sctp/sysctl.c b/net/sctp/sysctl.c
index 633cd17..e2c679b 100644
--- a/net/sctp/sysctl.c
+++ b/net/sctp/sysctl.c
@@ -254,7 +254,7 @@ static struct ctl_table_header * sctp_sysctl_header;
 /* Sysctl registration.  */
 void sctp_sysctl_register(void)
 {
- sctp_sysctl_header = register_sysctl_table(sctp_root_table, 0);
+ sctp_sysctl_header = register_sysctl_table(sctp_root_table);
 }

 /* Sysctl deregistration.  */
diff --git a/net/sunrpc/sysctl.c b/net/sunrpc/sysctl.c
index 6a82ed2..df70c8c 100644
--- a/net/sunrpc/sysctl.c
+++ b/net/sunrpc/sysctl.c
@@ -36,7 +36,7 @@ void
 rpc_register_sysctl(void)
 {
  if (!sunrpc_table_header)
-  sunrpc_table_header = register_sysctl_table(sunrpc_table, 0);
+  sunrpc_table_header = register_sysctl_table(sunrpc_table);
 }

 void
diff --git a/net/sunrpc/xprtsock.c b/net/sunrpc/xprtsock.c
index 51964cf..a314c86 100644
```

```
--- a/net/sunrpc/xprtsock.c
+++ b/net/sunrpc/xprtsock.c
@@ -1630,7 +1630,7 @@ int init_socket_xprt(void)
 {
 #ifdef RPC_DEBUG
  if (!sunrpc_table_header)
-  sunrpc_table_header = register_sysctl_table(sunrpc_table, 0);
+  sunrpc_table_header = register_sysctl_table(sunrpc_table);
 #endif

  return 0;
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c
index 690ffa5..eb0bd57 100644
--- a/net/unix/sysctl_net_unix.c
+++ b/net/unix/sysctl_net_unix.c
@@ -50,7 +50,7 @@ static struct ctl_table_header * unix_sysctl_header;

 void unix_sysctl_register(void)
 {
- unix_sysctl_header = register_sysctl_table(unix_root_table, 0);
+ unix_sysctl_header = register_sysctl_table(unix_root_table);
 }

 void unix_sysctl_unregister(void)
diff --git a/net/x25/sysctl_net_x25.c b/net/x25/sysctl_net_x25.c
index 94aff67..324ae1c 100644
--- a/net/x25/sysctl_net_x25.c
+++ b/net/x25/sysctl_net_x25.c
@@ -98,7 +98,7 @@ static struct ctl_table x25_root_table[] = {

 void __init x25_register_sysctl(void)
 {
- x25_table_header = register_sysctl_table(x25_root_table, 0);
+ x25_table_header = register_sysctl_table(x25_root_table);
 }

 void x25_unregister_sysctl(void)
--
1.4.4.1.g278f
```

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

Subject: [PATCH 56/59] sysctl: factor out sysctl_head_next from do_sysctl

Posted by ebiederm on Tue, 16 Jan 2007 16:40:01 GMT
View Forum Message <> Reply to Message

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

The current logic to walk through the list of sysctl table
headers is slightly painful and implement in a way it cannot
be used by code outside sysctl.c

I am in the process of implementing a version of the sysctl
proc support that instead of using the proc generic non-caching
monster, just uses the existing sysctl data structure as
backing store for building the dcache entries and for doing
directory reads. To use the existing data structures however
I need a way to get at them.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/linux/sysctl.h | 4 +++
 kernel/sysctl.c | 57 ++++++++++++++++++++++++++++++++++++----------
 2 files changed, 46 insertions(+), 15 deletions(-)

diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 6113f3b..81ee9ea 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -923,6 +923,10 @@ enum
 #ifdef __KERNEL__
 #include <linux/list.h>

+/* For the /proc/sys support */
+extern struct ctl_table_header *sysctl_head_next(struct ctl_table_header *prev);
+extern void sysctl_head_finish(struct ctl_table_header *prev);
+
 extern void sysctl_init(void);

 typedef struct ctl_table ctl_table;
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index 5beee1f..ca2831a 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -1066,6 +1066,42 @@ static void start_unregistering(struct ctl_table_header *p)
  list_del_init(&p->ctl_entry);
 }

+void sysctl_head_finish(struct ctl_table_header *head)
+{
+ if (!head)
+ return;

```
+ spin_lock(&sysctl_lock);
+ unuse_table(head);
+ spin_unlock(&sysctl_lock);
+}
+
+struct ctl_table_header *sysctl_head_next(struct ctl_table_header *prev)
+{
+ struct ctl_table_header *head;
+ struct list_head *tmp;
+ spin_lock(&sysctl_lock);
+ if (prev) {
+  tmp = &prev->ctl_entry;
+  unuse_table(prev);
+  goto next;
+ }
+ tmp = &root_table_header.ctl_entry;
+ for (;;) {
+  head = list_entry(tmp, struct ctl_table_header, ctl_entry);
+
+  if (!use_table(head))
+   goto next;
+  spin_unlock(&sysctl_lock);
+  return head;
+ next:
+  tmp = tmp->next;
+  if (tmp == &root_table_header.ctl_entry)
+   break;
+ }
+ spin_unlock(&sysctl_lock);
+ return NULL;
+}
+
 void __init sysctl_init(void)
 {
 #ifdef CONFIG_PROC_SYSCTL
@@ -1077,6 +1113,7 @@ void __init sysctl_init(void)
 int do_sysctl(int __user *name, int nlen, void __user *oldval, size_t __user *oldlenp,
         void __user *newval, size_t newlen)
 {
+ struct ctl_table_header *head;
  struct list_head *tmp;
  int error = -ENOTDIR;

@@ -1087,26 +1124,16 @@ int do_sysctl(int __user *name, int nlen, void __user *oldval, size_t
__user *ol
   if (!oldlenp || get_user(old_len, oldlenp))
    return -EFAULT;
  }
```

```
- spin_lock(&sysctl_lock);
- tmp = &root_table_header.ctl_entry;
- do {
-  struct ctl_table_header *head =
-   list_entry(tmp, struct ctl_table_header, ctl_entry);

-  if (!use_table(head))
-   continue;
-
-  spin_unlock(&sysctl_lock);
+ for (head = sysctl_head_next(NULL); head; head = sysctl_head_next(head)) {

   error = parse_table(name, nlen, oldval, oldlenp,
     newval, newlen, head->ctl_table);
-
-  spin_lock(&sysctl_lock);
-  unuse_table(head);
-  if (error != -ENOTDIR)
+  if (error != -ENOTDIR) {
+   sysctl_head_finish(head);
    break;
- } while ((tmp = tmp->next) != &root_table_header.ctl_entry);
- spin_unlock(&sysctl_lock);
+  }
+ }
  return error;
 }

--
1.4.4.1.g278f
```

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

Subject: [PATCH 57/59] sysctl: allow sysctl_perm to be called from outside of sysctl.c
Posted by ebiederm on Tue, 16 Jan 2007 16:40:02 GMT
View Forum Message <> Reply to Message

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/linux/sysctl.h | 2 ++
 kernel/sysctl.c | 10 +++++++---

```
 2 files changed, 7 insertions(+), 5 deletions(-)

diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 81ee9ea..20c23b5 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -924,8 +924,10 @@ enum
 #include <linux/list.h>

 /* For the /proc/sys support */
+struct ctl_table;
 extern struct ctl_table_header *sysctl_head_next(struct ctl_table_header *prev);
 extern void sysctl_head_finish(struct ctl_table_header *prev);
+extern int sysctl_perm(struct ctl_table *table, int op);

 extern void sysctl_init(void);

diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index ca2831a..ec5e4a1 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -1154,7 +1154,7 @@ asmlinkage long sys_sysctl(struct __sysctl_args __user *args)
 #endif /* CONFIG_SYSCTL_SYSCALL */

 /*
- * ctl_perm does NOT grant the superuser all rights automatically, because
+ * sysctl_perm does NOT grant the superuser all rights automatically, because
  * some sysctl variables are readonly even to root.
  */

@@ -1169,7 +1169,7 @@ static int test_perm(int mode, int op)
  return -EACCES;
 }

-static inline int ctl_perm(ctl_table *table, int op)
+int sysctl_perm(ctl_table *table, int op)
 {
  int error;
  error = security_sysctl(table, op);
@@ -1196,7 +1196,7 @@ repeat:
    if (n == table->ctl_name) {
     int error;
     if (table->child) {
-     if (ctl_perm(table, 001))
+     if (sysctl_perm(table, 001))
       return -EPERM;
      name++;
      nlen--;
```

```
@@ -1225,7 +1225,7 @@ int do_sysctl_strategy (ctl_table *table,
   op |= 004;
  if (newval)
   op |= 002;
- if (ctl_perm(table, op))
+ if (sysctl_perm(table, op))
   return -EPERM;

  if (table->strategy) {
@@ -1495,7 +1495,7 @@ static ssize_t do_rw_proc(int write, struct file * file, char __user * buf,
    goto out;
   error = -EPERM;
   op = (write ? 002 : 004);
-  if (ctl_perm(table, op))
+  if (sysctl_perm(table, op))
    goto out;

   /* careful: calling conventions are nasty here */
--
1.4.4.1.g278f
```

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

Subject: [PATCH 58/59] sysctl: Reimplement the sysctl proc support
Posted by ebiederm on Tue, 16 Jan 2007 16:40:03 GMT
View Forum Message <> Reply to Message

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

With this change the sysctl inodes can be cached and
nothing needs to be done when removing a sysctl table.

For a costk of 2K code we will save about 4K of static tables
(when we remove de from ctl_table) and 70K in proc_dir_entries
that we will not allocate, or about half that on a 32bit arch.

The speed feels about the same, even though we can now cache
the sysctl dentries :(

We get the core advantage that we don't need to have a
1 to 1 mapping between ctl table entries and proc files.
Making it possible to have /proc/sys vary depending on
the namespace you are in. The currently merged namespaces
don't have an issue here but the network namespace under

/proc/sys/net needs to have different directories depending
on which network adapters are visible. By simply being
a cache different directories being visible depending
on who you are is trivial to implement.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 fs/proc/Makefile | 2 +-
 fs/proc/inode.c | 1 +
 fs/proc/internal.h | 2 +
 fs/proc/proc_sysctl.c | 477 ++++++

```
+extern int proc_sys_init(void);
+
 struct vmalloc_info {
  unsigned long used;
  unsigned long largest_chunk;
diff --git a/fs/proc/proc_sysctl.c b/fs/proc/proc_sysctl.c
new file mode 100644
index 0000000..08a2e66
--- /dev/null
+++ b/fs/proc/proc_sysctl.c
@@ -0,0 +1,477 @@
+/*
+ * /proc/sys support
+ */
+
+#include <linux/sysctl.h>
+#include <linux/proc_fs.h>
+#include <linux/security.h>
+#include "internal.h"
+
+static struct dentry_operations proc_sys_dentry_operations;
+static const struct file_operations proc_sys_file_operations;
+static struct inode_operations proc_sys_inode_operations;
+
+static void proc_sys_refresh_inode(struct inode *inode, struct ctl_table *table)
+{
+ /* Refresh the cached information bits in the inode */
+ if (table) {
+  inode->i_uid = 0;
+  inode->i_gid = 0;
+  inode->i_mode = table->mode;
+  if (table->proc_handler) {
+   inode->i_mode |= S_IFREG;
+   inode->i_nlink = 1;
+  } else {
+   inode->i_mode |= S_IFDIR;
+   inode->i_nlink = 0; /* It is too hard to figure out */
+  }
+ }
+}
+
+static struct inode *proc_sys_make_inode(struct inode *dir, struct ctl_table *table)
+{
+ struct inode *inode;
+ struct proc_inode *dir_ei, *ei;
+ int depth;
+
```

```
+ inode = new_inode(dir->i_sb);
+ if (!inode)
+  goto out;
+
+ /* A directory is always one deeper than it's parent */
+ dir_ei = PROC_I(dir);
+ depth = dir_ei->fd + 1;
+
+ ei = PROC_I(inode);
+ ei->fd = depth;
+ inode->i_mtime = inode->i_atime = inode->i_ctime = CURRENT_TIME;
+ inode->i_op = &proc_sys_inode_operations;
+ inode->i_fop = &proc_sys_file_operations;
+ proc_sys_refresh_inode(inode, table);
+out:
+ return inode;
+}
+
+static struct dentry *proc_sys_ancestor(struct dentry *dentry, int depth)
+{
+ for (;;) {
+  struct proc_inode *ei;
+
+  ei = PROC_I(dentry->d_inode);
+  if (ei->fd == depth)
+   break; /* found */
+
+  dentry = dentry->d_parent;
+ }
+ return dentry;
+}
+
+static struct ctl_table *proc_sys_lookup_table_one(struct ctl_table *table,
+       struct qstr *name)
+{
+ int len;
+ for ( ; table->ctl_name || table->procname; table++) {
+
+  if (!table->procname)
+   continue;
+
+  len = strlen(table->procname);
+  if (len != name->len)
+   continue;
+
+  if (memcmp(table->procname, name->name, len) != 0)
+   continue;
+
```

```
+  /* I have a match */
+  return table;
+ }
+ return NULL;
+}
+
+static struct ctl_table *proc_sys_lookup_table(struct dentry *dentry,
+       struct ctl_table *table)
+{
+ struct dentry *ancestor;
+ struct proc_inode *ei;
+ int depth, i;
+
+ ei = PROC_I(dentry->d_inode);
+ depth = ei->fd;
+
+ if (depth == 0)
+  return table;
+
+ for (i = 1; table && (i <= depth); i++) {
+  ancestor = proc_sys_ancestor(dentry, i);
+  table = proc_sys_lookup_table_one(table, &ancestor->d_name);
+  if (table)
+   table = table->child;
+ }
+ return table;
+
+}
+static struct ctl_table *proc_sys_lookup_entry(struct dentry *dparent,
+       struct qstr *name,
+       struct ctl_table *table)
+{
+ table = proc_sys_lookup_table(dparent, table);
+ if (table)
+  table = proc_sys_lookup_table_one(table, name);
+ return table;
+}
+
+static struct ctl_table *do_proc_sys_lookup(struct dentry *parent,
+       struct qstr *name,
+       struct ctl_table_header **ptr)
+{
+ struct ctl_table_header *head;
+ struct ctl_table *table;
+
+ for (head = sysctl_head_next(NULL); head; head = sysctl_head_next(head)) {
+  table = proc_sys_lookup_entry(parent, name, head->ctl_table);
+  if (table)
```

```
+   break;
+ }
+ *ptr = head;
+ return table;
+}
+
+static struct dentry *proc_sys_lookup(struct inode *dir, struct dentry *dentry,
+     struct nameidata *nd)
+{
+ struct ctl_table_header *head;
+ struct inode *inode;
+ struct dentry *err;
+ struct ctl_table *table;
+
+ err = ERR_PTR(-ENOENT);
+ table = do_proc_sys_lookup(dentry->d_parent, &dentry->d_name, &head);
+ if (!table)
+  goto out;
+
+ err = ERR_PTR(-ENOMEM);
+ inode = proc_sys_make_inode(dir, table);
+ if (!inode)
+  goto out;
+
+ err = NULL;
+ dentry->d_op = &proc_sys_dentry_operations;
+ d_add(dentry, inode);
+
+out:
+ sysctl_head_finish(head);
+ return err;
+}
+
+static ssize_t proc_sys_read(struct file *filp, char __user *buf,
+    size_t count, loff_t *ppos)
+{
+ struct dentry *dentry = filp->f_dentry;
+ struct ctl_table_header *head;
+ struct ctl_table *table;
+ ssize_t error, res;
+
+ table = do_proc_sys_lookup(dentry->d_parent, &dentry->d_name, &head);
+ /* Has the sysctl entry disappeared on us? */
+ error = -ENOENT;
+ if (!table)
+  goto out;
+
+ /* Has the sysctl entry been replaced by a directory? */
```

```
+ error = -EISDIR;
+ if (!table->proc_handler)
+  goto out;
+
+ /*
+  * At this point we know that the sysctl was not unregistered
+  * and won't be until we finish.
+  */
+ error = -EPERM;
+ if (sysctl_perm(table, MAY_READ))
+  goto out;
+
+ /* careful: calling conventions are nasty here */
+ res = count;
+ error = table->proc_handler(table, 0, filp, buf, &res, ppos);
+ if (!error)
+  error = res;
+out:
+ sysctl_head_finish(head);
+
+ return error;
+}
+
+static ssize_t proc_sys_write(struct file *filp, const char __user *buf,
+    size_t count, loff_t *ppos)
+{
+ struct dentry *dentry = filp->f_dentry;
+ struct ctl_table_header *head;
+ struct ctl_table *table;
+ ssize_t error, res;
+
+ table = do_proc_sys_lookup(dentry->d_parent, &dentry->d_name, &head);
+ /* Has the sysctl entry disappeared on us? */
+ error = -ENOENT;
+ if (!table)
+  goto out;
+
+ /* Has the sysctl entry been replaced by a directory? */
+ error = -EISDIR;
+ if (!table->proc_handler)
+  goto out;
+
+ /*
+  * At this point we know that the sysctl was not unregistered
+  * and won't be until we finish.
+  */
+ error = -EPERM;
+ if (sysctl_perm(table, MAY_WRITE))
```

```
+  goto out;
+
+ /* careful: calling conventions are nasty here */
+ res = count;
+ error = table->proc_handler(table, 1, filp, buf, &res, ppos);
+ if (!error)
+  error = res;
+out:
+ sysctl_head_finish(head);
+
+ return error;
+}
+
+
+static int proc_sys_fill_cache(struct file *filp, void *dirent,
+    filldir_t filldir, struct ctl_table *table)
+{
+ struct ctl_table_header *head;
+ struct ctl_table *child_table = NULL;
+ struct dentry *child, *dir = filp->f_path.dentry;
+ struct inode *inode;
+ struct qstr qname;
+ ino_t ino = 0;
+ unsigned type = DT_UNKNOWN;
+ int ret;
+
+ qname.name = table->procname;
+ qname.len  = strlen(table->procname);
+ qname.hash = full_name_hash(qname.name, qname.len);
+
+ /* Suppress duplicates.
+  * Only fill a directory entry if it is the value that
+  * an ordinary lookup of that name returns.  Hide all
+  * others.
+  *
+  * If we ever cache this translation in the dcache
+  * I should do a dcache lookup first.  But for now
+  * it is just simpler not to.
+  */
+ ret = 0;
+ child_table = do_proc_sys_lookup(dir, &qname, &head);
+ sysctl_head_finish(head);
+ if (child_table != table)
+  return 0;
+
+ child = d_lookup(dir, &qname);
+ if (!child) {
+  struct dentry *new;
```

```
+  new = d_alloc(dir, &qname);
+  if (new) {
+   inode = proc_sys_make_inode(dir->d_inode, table);
+   if (!inode)
+    child = ERR_PTR(-ENOMEM);
+   else {
+    new->d_op = &proc_sys_dentry_operations;
+    d_add(new, inode);
+   }
+   if (child)
+    dput(new);
+   else
+    child = new;
+  }
+ }
+ if (!child || IS_ERR(child) || !child->d_inode)
+  goto end_instantiate;
+ inode = child->d_inode;
+ if (inode) {
+  ino  = inode->i_ino;
+  type = inode->i_mode >> 12;
+ }
+ dput(child);
+end_instantiate:
+ if (!ino)
+  ino= find_inode_number(dir, &qname);
+ if (!ino)
+  ino = 1;
+ return filldir(dent, qname.name, qname.len, filp->f_pos, ino, type);
+}
+
+static int proc_sys_readdir(struct file *filp, void *dirent, filldir_t filldir)
+{
+ struct dentry *dentry = filp->f_dentry;
+ struct inode *inode = dentry->d_inode;
+ struct ctl_table_header *head = NULL;
+ struct ctl_table *table;
+ unsigned long pos;
+ int ret;
+
+ ret = -ENOTDIR;
+ if (!S_ISDIR(inode->i_mode))
+  goto out;
+
+ ret = 0;
+ switch(filp->f_pos) {
+ case 0:
+  if (filldir(dirent, ".", 1, filp->f_pos, inode->i_ino, DT_DIR) < 0)
```

```
+   goto out;
+  filp->f_pos++;
+  /* fall through */
+ case 1:
+  if (filldir(dirent, "..", 2, filp->f_pos, parent_ino(dentry), DT_DIR) < 0)
+   goto out;
+  filp->f_pos++;
+  /* fall through */
+ default:
+  pos = 2;
+  break;
+ }
+
+ /* - Find each instance of the directory
+  * - Read all entries in each instance
+  * - Before returning an entry to user space lookup the entry
+  *   by name and if I find a different entry don't return
+  *   this one because it means it is a buried dup.
+  * For sysctl this should only happen for directory entries.
+  */
+ for (head = sysctl_head_next(NULL); head; head = sysctl_head_next(head)) {
+  table = proc_sys_lookup_table(dentry, head->ctl_table);
+
+  if (!table)
+   continue;
+
+  for (; table->ctl_name || table->procname; table++, pos++) {
+   /* Can't do anything without a proc name */
+   if (!table->procname)
+    continue;
+
+   if (pos < filp->f_pos)
+    continue;
+
+   if (proc_sys_fill_cache(filp, dirent, filldir, table) < 0)
+    goto out;
+   filp->f_pos = pos + 1;
+  }
+ }
+ ret = 1;
+out:
+ sysctl_head_finish(head);
+ return ret;
+}
+
+static int proc_sys_permission(struct inode *inode, int mask, struct nameidata *nd)
+{
+ /*
```

```
+  * sysctl entries that are not writeable,
+  * are _NOT_ writeable, capabilities or not.
+  */
+ struct ctl_table_header *head;
+ struct ctl_table *table;
+ struct dentry *dentry;
+ int mode;
+ int depth;
+ int error;
+
+ head = NULL;
+ depth = PROC_I(inode)->fd;
+
+ /* First check the cached permissions, in case we don't have
+  * enough information to lookup the sysctl table entry.
+  */
+ error = -EACCES;
+ mode = inode->i_mode;
+
+ if (current->euid == 0)
+  mode >>= 6;
+ else if (in_group_p(0))
+  mode >>= 3;
+
+ if ((mode & mask & (MAY_READ|MAY_WRITE|MAY_EXEC)) == mask)
+  error = 0;
+
+ /* If we can't get a sysctl table entry the permission
+  * checks on the cached mode will have to be enough.
+  */
+ if (!nd || !depth)
+  goto out;
+
+ dentry = nd->dentry;
+ table = do_proc_sys_lookup(dentry->d_parent, &dentry->d_name, &head);
+
+ /* If the entry does not exist deny permission */
+ error = -EACCES;
+ if (!table)
+  goto out;
+
+ /* Use the permissions on the sysctl table entry */
+ error = sysctl_perm(table, mask);
+out:
+ sysctl_head_finish(head);
+ return error;
+}
+
```

```
+static int proc_sys_setattr(struct dentry *dentry, struct iattr *attr)
+{
+ struct inode *inode = dentry->d_inode;
+ int error;
+
+ if (attr->ia_valid & (ATTR_MODE | ATTR_UID | ATTR_GID))
+  return -EPERM;
+
+ error = inode_change_ok(inode, attr);
+ if (!error) {
+  error = security_inode_setattr(dentry, attr);
+  if (!error)
+   error = inode_setattr(inode, attr);
+ }
+
+ return error;
+}
+
+/* I'm lazy and don't distinguish between files and directories,
+ * until access time.
+ */
+static const struct file_operations proc_sys_file_operations = {
+ .read  = proc_sys_read,
+ .write  = proc_sys_write,
+ .readdir = proc_sys_readdir,
+};
+
+static struct inode_operations proc_sys_inode_operations = {
+ .lookup  = proc_sys_lookup,
+ .permission = proc_sys_permission,
+ .setattr = proc_sys_setattr,
+};
+
+static int proc_sys_revalidate(struct dentry *dentry, struct nameidata *nd)
+{
+ struct ctl_table_header *head;
+ struct ctl_table *table;
+ table = do_proc_sys_lookup(dentry->d_parent, &dentry->d_name, &head);
+ proc_sys_refresh_inode(dentry->d_inode, table);
+ sysctl_head_finish(head);
+ return !!table;
+}
+
+static struct dentry_operations proc_sys_dentry_operations = {
+ .d_revalidate = proc_sys_revalidate,
+};
+
+struct proc_dir_entry *proc_sys_root;
```

```
+
+int proc_sys_init(void)
+{
+ proc_sys_root = proc_mkdir("sys", NULL);
+ proc_sys_root->proc_iops = &proc_sys_inode_operations;
+ proc_sys_root->proc_fops = &proc_sys_file_operations;
+ proc_sys_root->nlink = 0;
+ return 0;
+}
diff --git a/fs/proc/root.c b/fs/proc/root.c
index 8059e92..4d42406 100644
--- a/fs/proc/root.c
+++ b/fs/proc/root.c
@@ -23,10 +23,6 @@

 struct proc_dir_entry *proc_net, *proc_net_stat, *proc_bus, *proc_root_fs, *proc_root_driver;

-#ifdef CONFIG_SYSCTL
-struct proc_dir_entry *proc_sys_root;
-#endif
-
 static int proc_get_sb(struct file_system_type *fs_type,
  int flags, const char *dev_name, void *data, struct vfsmount *mnt)
 {
@@ -71,9 +67,6 @@ void __init proc_root_init(void)
 #ifdef CONFIG_SYSVIPC
  proc_mkdir("sysvipc", NULL);
 #endif
-#ifdef CONFIG_SYSCTL
- proc_sys_root = proc_mkdir("sys", NULL);
-#endif
  proc_root_fs = proc_mkdir("fs", NULL);
  proc_root_driver = proc_mkdir("driver", NULL);
  proc_mkdir("fs/nfsd", NULL); /* somewhere for the nfsd filesystem to be mounted */
@@ -86,6 +79,9 @@ void __init proc_root_init(void)
  proc_device_tree_init();
 #endif
  proc_bus = proc_mkdir("bus", NULL);
+#ifdef CONFIG_SYSCTL
+ proc_sys_init();
+#endif
 }

 static int proc_root_getattr(struct vfsmount *mnt, struct dentry *dentry, struct kstat *stat
diff --git a/init/main.c b/init/main.c
index 8af5c6e..7926e5d 100644
--- a/init/main.c
+++ b/init/main.c
```

```
@@ -86,7 +86,6 @@ extern void init_IRQ(void);
 extern void fork_init(unsigned long);
 extern void mca_init(void);
 extern void sbus_init(void);
-extern void sysctl_init(void);
 extern void signals_init(void);
 extern void pidhash_init(void);
 extern void pidmap_init(void);
@@ -688,9 +687,6 @@ static void __init do_basic_setup(void)
  usermodehelper_init();
  driver_init();

-#ifdef CONFIG_SYSCTL
- sysctl_init();
-#endif
 #ifdef CONFIG_PROC_FS
  init_irq_proc();
 #endif
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index ec5e4a1..4b45bdb 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -159,26 +159,6 @@ int sysctl_legacy_va_layout;



-/* /proc declarations: */
-
-#ifdef CONFIG_PROC_SYSCTL
-
-static ssize_t proc_readsys(struct file *, char __user *, size_t, loff_t *);
-static ssize_t proc_writesys(struct file *, const char __user *, size_t, loff_t *);
-static int proc_opensys(struct inode *, struct file *);
-
-const struct file_operations proc_sys_file_operations = {
- .open  = proc_opensys,
- .read  = proc_readsys,
- .write  = proc_writesys,
-};
-
-extern struct proc_dir_entry *proc_sys_root;
-
-static void register_proc_table(ctl_table *, struct proc_dir_entry *, void *);
-static void unregister_proc_table(ctl_table *, struct proc_dir_entry *);
-#endif
-
 /* The default sysctl tables: */
```

```
 static ctl_table root_table[] = {
@@ -1102,13 +1082,6 @@ struct ctl_table_header *sysctl_head_next(struct ctl_table_header
*prev)
  return NULL;
 }

-void __init sysctl_init(void)
-{
-#ifdef CONFIG_PROC_SYSCTL
- register_proc_table(root_table, proc_sys_root, &root_table_header);
-#endif
-}
-
 #ifdef CONFIG_SYSCTL_SYSCALL
 int do_sysctl(int __user *name, int nlen, void __user *oldval, size_t __user *oldlenp,
         void __user *newval, size_t newlen)
@@ -1345,9 +1318,6 @@ struct ctl_table_header *register_sysctl_table(ctl_table * table)
  spin_lock(&sysctl_lock);
  list_add_tail(&tmp->ctl_entry, &root_table_header.ctl_entry);
  spin_unlock(&sysctl_lock);
-#ifdef CONFIG_PROC_SYSCTL
- register_proc_table(table, proc_sys_root, tmp);
-#endif
  return tmp;
 }

@@ -1363,9 +1333,6 @@ void unregister_sysctl_table(struct ctl_table_header * header)
  might_sleep();
  spin_lock(&sysctl_lock);
  start_unregistering(header);
-#ifdef CONFIG_PROC_SYSCTL
- unregister_proc_table(header->ctl_table, proc_sys_root);
-#endif
  spin_unlock(&sysctl_lock);
  kfree(header);
 }
@@ -1389,155 +1356,6 @@ void unregister_sysctl_table(struct ctl_table_header * table)

 #ifdef CONFIG_PROC_SYSCTL

-/* Scan the sysctl entries in table and add them all into /proc */
-static void register_proc_table(ctl_table * table, struct proc_dir_entry *root, void *set)
-{
- struct proc_dir_entry *de;
- int len;
- mode_t mode;
-
- for (; table->ctl_name || table->procname; table++) {
```

```
-  /* Can't do anything without a proc name. */
-  if (!table->procname)
-   continue;
-  /* Maybe we can't do anything with it... */
-  if (!table->proc_handler && !table->child) {
-   printk(KERN_WARNING "SYSCTL: Can't register %s\n",
-    table->procname);
-   continue;
-  }
-
-  len = strlen(table->procname);
-  mode = table->mode;
-
-  de = NULL;
-  if (table->proc_handler)
-   mode |= S_IFREG;
-  else {
-   mode |= S_IFDIR;
-   for (de = root->subdir; de; de = de->next) {
-    if (proc_match(len, table->procname, de))
-     break;
-   }
-   /* If the subdir exists already, de is non-NULL */
-  }
-
-  if (!de) {
-   de = create_proc_entry(table->procname, mode, root);
-   if (!de)
-    continue;
-   de->set = set;
-   de->data = (void *) table;
-   if (table->proc_handler)
-    de->proc_fops = &proc_sys_file_operations;
-  }
-  table->de = de;
-  if (de->mode & S_IFDIR)
-   register_proc_table(table->child, de, set);
- }
-}
-
-/*
- * Unregister a /proc sysctl table and any subdirectories.
- */
-static void unregister_proc_table(ctl_table * table, struct proc_dir_entry *root)
-{
- struct proc_dir_entry *de;
- for (; table->ctl_name || table->procname; table++) {
-  if (!(de = table->de))
```

```
-   continue;
-  if (de->mode & S_IFDIR) {
-   if (!table->child) {
-    printk (KERN_ALERT "Help - malformed sysctl tree on free\n");
-    continue;
-   }
-   unregister_proc_table(table->child, de);
-
-   /* Don't unregister directories which still have entries.. */
-   if (de->subdir)
-    continue;
-  }
-
-  /*
-   * In any case, mark the entry as goner; we'll keep it
-   * around if it's busy, but we'll know to do nothing with
-   * its fields.  We are under sysctl_lock here.
-   */
-  de->data = NULL;
-
-  /* Don't unregister proc entries that are still being used.. */
-  if (atomic_read(&de->count))
-   continue;
-
-  table->de = NULL;
-  remove_proc_entry(table->procname, root);
- }
-}
-
-static ssize_t do_rw_proc(int write, struct file * file, char __user * buf,
-     size_t count, loff_t *ppos)
-{
- int op;
- struct proc_dir_entry *de = PDE(file->f_path.dentry->d_inode);
- struct ctl_table *table;
- size_t res;
- ssize_t error = -ENOTDIR;
-
- spin_lock(&sysctl_lock);
- if (de && de->data && use_table(de->set)) {
-  /*
-   * at that point we know that sysctl was not unregistered
-   * and won't be until we finish
-   */
-  spin_unlock(&sysctl_lock);
-  table = (struct ctl_table *) de->data;
-  if (!table || !table->proc_handler)
-   goto out;
```

```
-  error = -EPERM;
-  op = (write ? 002 : 004);
-  if (sysctl_perm(table, op))
-   goto out;
-
-  /* careful: calling conventions are nasty here */
-  res = count;
-  error = (*table->proc_handler)(table, write, file,
-      buf, &res, ppos);
-  if (!error)
-   error = res;
- out:
-  spin_lock(&sysctl_lock);
-  unuse_table(de->set);
- }
- spin_unlock(&sysctl_lock);
- return error;
-}
-
-static int proc_opensys(struct inode *inode, struct file *file)
-{
- if (file->f_mode & FMODE_WRITE) {
-  /*
-   * sysctl entries that are not writable,
-   * are _NOT_ writable, capabilities or not.
-   */
-  if (!(inode->i_mode & S_IWUSR))
-   return -EPERM;
- }
-
- return 0;
-}
-
-static ssize_t proc_readsys(struct file * file, char __user * buf,
-       size_t count, loff_t *ppos)
-{
- return do_rw_proc(0, file, buf, count, ppos);
-}
-
-static ssize_t proc_writesys(struct file * file, const char __user * buf,
-        size_t count, loff_t *ppos)
-{
- return do_rw_proc(1, file, (char __user *) buf, count, ppos);
-}
-
 static int _proc_do_string(void* data, int maxlen, int write,
       struct file *filp, void __user *buffer,
       size_t *lenp, loff_t *ppos)
```

--
1.4.4.1.g278f

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

Subject: [PATCH 59/59] sysctl: Remove the proc_dir_entry member for the sysctl tables.
Posted by ebiederm on Tue, 16 Jan 2007 16:40:04 GMT
View Forum Message <> Reply to Message

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

It isn't needed anymore, all of the users are gone, and all of the
ctl_table initializers have been converted to use explicit names
of the fields they are initializing.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 include/linux/sysctl.h | 1 -
 net/decnet/dn_dev.c | 5 -----
 net/ipv4/devinet.c | 5 -----
 net/ipv6/addrconf.c | 5 -----
 4 files changed, 0 insertions(+), 16 deletions(-)

diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
index 20c23b5..8c2fab5 100644
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -1025,7 +1025,6 @@ struct ctl_table
  ctl_table *child;
  proc_handler *proc_handler; /* Callback for text formatting */
  ctl_handler *strategy; /* Callback function for all r/w */
- struct proc_dir_entry *de; /* /proc control block */
  void *extra1;
  void *extra2;
 };
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index baaa02e..324eb47 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -261,7 +261,6 @@ static void dn_dev_sysctl_register(struct net_device *dev, struct
dn_dev_parms *
  for(i = 0; i < ARRAY_SIZE(t->dn_dev_vars) - 1; i++) {
   long offset = (long)t->dn_dev_vars[i].data;

```
   t->dn_dev_vars[i].data = ((char *)parms) + offset;
-  t->dn_dev_vars[i].de = NULL;
   }

   if (dev) {
@@ -273,13 +272,9 @@ static void dn_dev_sysctl_register(struct net_device *dev, struct
dn_dev_parms *
   }

   t->dn_dev_dev[0].child = t->dn_dev_vars;
- t->dn_dev_dev[0].de = NULL;
   t->dn_dev_conf_dir[0].child = t->dn_dev_dev;
- t->dn_dev_conf_dir[0].de = NULL;
   t->dn_dev_proto_dir[0].child = t->dn_dev_conf_dir;
- t->dn_dev_proto_dir[0].de = NULL;
   t->dn_dev_root_dir[0].child = t->dn_dev_proto_dir;
- t->dn_dev_root_dir[0].de = NULL;
   t->dn_dev_vars[0].extra1 = (void *)dev;

   t->sysctl_header = register_sysctl_table(t->dn_dev_root_dir);
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index b731a0c..8cfcc78 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1573,7 +1573,6 @@ static void devinet_sysctl_register(struct in_device *in_dev,
    return;
   for (i = 0; i < ARRAY_SIZE(t->devinet_vars) - 1; i++) {
    t->devinet_vars[i].data += (char *)p - (char *)&ipv4_devconf;
-   t->devinet_vars[i].de = NULL;
   }

   if (dev) {
@@ -1595,13 +1594,9 @@ static void devinet_sysctl_register(struct in_device *in_dev,

   t->devinet_dev[0].procname    = dev_name;
   t->devinet_dev[0].child       = t->devinet_vars;
- t->devinet_dev[0].de       = NULL;
   t->devinet_conf_dir[0].child  = t->devinet_dev;
- t->devinet_conf_dir[0].de     = NULL;
   t->devinet_proto_dir[0].child = t->devinet_conf_dir;
- t->devinet_proto_dir[0].de    = NULL;
   t->devinet_root_dir[0].child  = t->devinet_proto_dir;
- t->devinet_root_dir[0].de     = NULL;

   t->sysctl_header = register_sysctl_table(t->devinet_root_dir);
   if (!t->sysctl_header)
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 791aaba..6aded83 100644
```

```
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -3973,7 +3973,6 @@ static void addrconf_sysctl_register(struct inet6_dev *idev, struct
ipv6_devconf
   return;
  for (i=0; t->addrconf_vars[i].data; i++) {
   t->addrconf_vars[i].data += (char*)p - (char*)&ipv6_devconf;
-  t->addrconf_vars[i].de = NULL;
   t->addrconf_vars[i].extra1 = idev; /* embedded; no ref */
  }
  if (dev) {
@@ -3996,13 +3995,9 @@ static void addrconf_sysctl_register(struct inet6_dev *idev, struct
ipv6_devconf
  t->addrconf_dev[0].procname = dev_name;

  t->addrconf_dev[0].child = t->addrconf_vars;
- t->addrconf_dev[0].de = NULL;
  t->addrconf_conf_dir[0].child = t->addrconf_dev;
- t->addrconf_conf_dir[0].de = NULL;
  t->addrconf_proto_dir[0].child = t->addrconf_conf_dir;
- t->addrconf_proto_dir[0].de = NULL;
  t->addrconf_root_dir[0].child = t->addrconf_proto_dir;
- t->addrconf_root_dir[0].de = NULL;

  t->sysctl_header = register_sysctl_table(t->addrconf_root_dir);
  if (t->sysctl_header == NULL)
--
1.4.4.1.g278f
```

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

Subject: Re: [PATCH 5/59] sysctl: rose remove unnecessary insert_at_head flag
Posted by Ralf Baechle on Tue, 16 Jan 2007 17:36:51 GMT
View Forum Message <> Reply to Message

On Tue, Jan 16, 2007 at 09:39:10AM -0700, Eric W. Biederman wrote:

Looks ok, for these:

Subject: [PATCH 5/59] sysctl: rose remove unnecessary insert_at_head flag
Subject: [PATCH 6/59] sysctl: netrom remove unnecessary insert_at_head flag
Subject: [PATCH 11/59] sysctl: ax25 remove unnecessary insert_at_head flag
Subject: [PATCH 30/59] sysctl: mips/au1000 Remove sys_sysctl support
Subject: [PATCH 31/59] sysctl: C99 convert the ctl_tables in arch/mips/au1000/common/power.c

Subject: [PATCH 32/59] sysctl: C99 convert arch/mips/lasat/sysctl.c and remove ABI breakage.
Subject: [PATCH 43/59] sysctl: Remove sys_sysctl support from drivers/char/rtc.c
Subject: [PATCH 55/59] sysctl: Remove insert_at_head from register_sysctl

Acked-by: Ralf Baechle <ralf@linux-mips.org>

Ralf

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

Subject: Re: [PATCH 20/59] sysctl: cdrom Don't set de->owner
Posted by James Bottomley on Tue, 16 Jan 2007 18:19:34 GMT
View Forum Message <> Reply to Message

On Tue, 2007-01-16 at 09:39 -0700, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

These three look OK:

[PATCH 15/59] sysctl: scsi remove unnecessary insert_at_head flag
[PATCH 19/59] sysctl: cdrom remove unnecessary insert_at_head flag
[PATCH 20/59] sysctl: cdrom Don't set de->owner

So you can add an ACK from me.

It would have been nice not to have 56 other irrelevant patches sprayed
over the list and into my inbox, though ...

James

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by hpa on Tue, 16 Jan 2007 18:35:03 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
>
>> I think it would be fair to say that if they're not in <linux/sysctl.h> they're

>> not architectural, but that doesn't resolve the counterpositive (are there
>> sysctls in <linux/sysctl.h> which aren't architectural? From the looks of it, I
>> would say yes.) Non-architectural sysctl numbers should not be exported to
>> userspace, and should eventually be rejected by sys_sysctl.
>
> This last bit doesn't make much sense. I believe you are saying all sysctl
> numbers should be per architecture.
>

With "architectural" I mean "guaranteed to be stable" (as opposed to
"incidental"). Sorry for the confusion.

-hpa

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by ebiederm on Tue, 16 Jan 2007 18:54:09 GMT
View Forum Message <> Reply to Message

"H. Peter Anvin" <hpa@zytor.com> writes:

> Eric W. Biederman wrote:
>>
>>> I think it would be fair to say that if they're not in <linux/sysctl.h>
> they're
>>> not architectural, but that doesn't resolve the counterpositive (are there
>>> sysctls in <linux/sysctl.h> which aren't architectural? From the looks of
> it, I
>>> would say yes.) Non-architectural sysctl numbers should not be exported to
>>> userspace, and should eventually be rejected by sys_sysctl.
>>
>> This last bit doesn't make much sense. I believe you are saying all sysctl
>> numbers should be per architecture.
>>
>
> With "architectural" I mean "guaranteed to be stable" (as opposed to
> "incidental"). Sorry for the confusion.

Ok. Then largely we are in agreement. To implement that the rule is simple.
If it isn't CTL_UNNUMBERED and the number is in Linus's tree, it is
our responsibility to never change the meaning of that number.

If a new sysctl entry is introduced it should be CTL_UNNUMBERED until
it reaches Linus's tree to avoid conflicts.

There is simply no point in having any kind of support for numbers
whose meanings can change.

Which is why I removed the few cases of binary number duplication I
found.

Eric

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by hpa on Tue, 16 Jan 2007 18:58:10 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
>>>
>> With "architectural" I mean "guaranteed to be stable" (as opposed to
>> "incidental"). Sorry for the confusion.
>
> Ok. Then largely we are in agreement. To implement that the rule is simple.
> If it isn't CTL_UNNUMBERED and the number is in Linus's tree, it is
> our responsibility to never change the meaning of that number.
>
> If a new sysctl entry is introduced it should be CTL_UNNUMBERED until
> it reaches Linus's tree to avoid conflicts.
>
> There is simply no point in having any kind of support for numbers
> whose meanings can change.
>
> Which is why I removed the few cases of binary number duplication I
> found.
>

Agreed. *Furthermore*, if the number isn't in <linux/sysctl.h> it
shouldn't exist anywhere else, either.

-hpa

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by ebiederm on Tue, 16 Jan 2007 19:03:36 GMT
View Forum Message <> Reply to Message

"H. Peter Anvin" <hpa@zytor.com> writes:

> Eric W. Biederman wrote:
>>>>
>>> With "architectural" I mean "guaranteed to be stable" (as opposed to
>>> "incidental"). Sorry for the confusion.
>>
>> Ok. Then largely we are in agreement. To implement that the rule is simple.
>> If it isn't CTL_UNNUMBERED and the number is in Linus's tree, it is
>> our responsibility to never change the meaning of that number.
>>
>> If a new sysctl entry is introduced it should be CTL_UNNUMBERED until
>> it reaches Linus's tree to avoid conflicts.
>>
>> There is simply no point in having any kind of support for numbers
>> whose meanings can change.
>>
>> Which is why I removed the few cases of binary number duplication I
>> found.
>>
>
> Agreed. *Furthermore*, if the number isn't in <linux/sysctl.h> it shouldn't
> exist anywhere else, either.

That would be a good habit. Feel free to send the patches to ensure that
is so.

I'm a practical fix it when it is in my way kind of guy ;)

Eric

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by hpa on Tue, 16 Jan 2007 19:15:05 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
>>
>> Agreed. *Furthermore*, if the number isn't in <linux/sysctl.h> it shouldn't
>> exist anywhere else, either.

>
> That would be a good habit. Feel free to send the patches to ensure that
> is so.
>
> I'm a practical fix it when it is in my way kind of guy ;)

That's fine. However, I am wondering if there are things in
<linux/sysctl.h> which really doesn't need architectural numbers, i.e.
which should be removed from the binary interface.

 -hpa

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by ebiederm on Tue, 16 Jan 2007 19:30:53 GMT
View Forum Message <> Reply to Message

"H. Peter Anvin" <hpa@zytor.com> writes:

> Eric W. Biederman wrote:
>>>
>>> Agreed. *Furthermore*, if the number isn't in <linux/sysctl.h> it shouldn't
>>> exist anywhere else, either.
>>
>> That would be a good habit. Feel free to send the patches to ensure that
>> is so.
>>
>> I'm a practical fix it when it is in my way kind of guy ;)
>
> That's fine. However, I am wondering if there are things in <linux/sysctl.h>
> which really doesn't need architectural numbers, i.e. which should be removed
> from the binary interface.

As a practical measure the 32bit hierarchical numeric namespace is extremely
sparsely populated. So even if there are things that we never intend to export
again it doesn't hurt to reserve the number for them so we don't confuse something
that thought the number actually meant something.

In the worst case we want to comment out the entry so we keep the number
reserved even if we don't use it.

To see if there are entries we are not currently using just requires
going through the loop.

<pseudo shell>
For DEFINE in sysctl.h ; do
  if ! grep -r DEFINE /usr/src/linux/ ; then
   echo DEFINE not used.
  fi
done
</pseudo shell>

Eric

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by David Howells on Tue, 16 Jan 2007 20:02:21 GMT
View Forum Message <> Reply to Message

The FRV bits look okay. I can't test them until I get back from Australia in
Feb.

David

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

Subject: Re: [PATCH 48/59] sysctl: Register the ocfs2 sysctl numbers
Posted by Mark Fasheh on Tue, 16 Jan 2007 20:37:10 GMT
View Forum Message <> Reply to Message

On Tue, Jan 16, 2007 at 09:39:53AM -0700, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> ocfs2 was did not have the binary number it uses under CTL_FS
> registered in sysctl.h. Register it to avoid future conflicts,
> and change the name of the definition to be in line with the
> rest of the sysctl numbers.

This looks good - ACK.
 --Mark

--
Mark Fasheh
Senior Software Developer, Oracle

mark.fasheh@oracle.com

_____

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

Subject: Re: [PATCH 37/59] sysctl: C99 convert arch/sh64/kernel/traps.c and remove ABI breakage.
Posted by Paul Mundt on Tue, 16 Jan 2007 22:07:08 GMT
View Forum Message <> Reply to Message

On Tue, Jan 16, 2007 at 09:39:42AM -0700, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> While doing the C99 conversion I notices that the top level sh64
> directory was using the binary number for CTL_KERN. That is a
> no-no so I removed the support for the sysctl binary interface
> only leaving sysctl /proc support.
>
> At least the sysctl tables were placed at the end of
> the list so user space did not see this mistake.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Looks good, thanks Eric.

Acked-by: Paul Mundt <lethal@linux-sh.org>

_____

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

Subject: Re: [PATCH 45/59] sysctl: C99 convert ctl_tables in drivers/parport/procfs.c
Posted by Ingo Oeser on Tue, 16 Jan 2007 22:15:43 GMT
View Forum Message <> Reply to Message

Hi Eric,

On Tuesday, 16. January 2007 17:39, Eric W. Biederman wrote:
> diff --git a/drivers/parport/procfs.c b/drivers/parport/procfs.c
> index 2e744a2..5337789 100644
> --- a/drivers/parport/procfs.c
> +++ b/drivers/parport/procfs.c
> @@ -263,50 +263,118 @@ struct parport_sysctl_table {

> + {
> + .ctl_name = DEV_PARPORT_BASE_ADDR,
> + .procname = "base-addr",
> + .data = NULL,
> + .maxlen = 0,
> + .mode = 0444,
> + .proc_handler = &do_hardware_base_addr
> + },

No need to initialize to zero or NULL. Just list any variable, which is NOT zero or NULL.

> + {
> + .ctl_name = DEV_PARPORT_AUTOPROBE + 1,
> + .procname = "autoprobe0",
> + .data = NULL,
> + .maxlen = 0,
> + .maxlen = 0444,
> + .proc_handler = &do_autoprobe
> + },

Typo here? .mode = 0444 makes mor sense.

Regards

Ingo Oeser

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

Subject: Re: [PATCH 31/59] sysctl: C99 convert the ctl_tables in arch/mips/au1000/common/power.c
Posted by Ingo Oeser on Tue, 16 Jan 2007 22:20:17 GMT
View Forum Message <> Reply to Message

Hi Eric,

On Tuesday, 16. January 2007 17:39, Eric W. Biederman wrote:
> diff --git a/arch/mips/au1000/common/power.c b/arch/mips/au1000/common/power.c
> index b531ab7..31256b8 100644
> --- a/arch/mips/au1000/common/power.c
> +++ b/arch/mips/au1000/common/power.c
> @@ -419,15 +419,41 @@ static int pm_do_freq(ctl_table * ctl, int write, struct file *file,

> + {
> + .ctl_name = CTL_UNNUMBERED,
> + .procname = "suspend",

> + .data = NULL,
> + .maxlen = 0,
> + .mode = 0600,
> + .proc_handler = &pm_do_suspend
> + },

No need for zero initialization for maxlen.

> + {
> + .ctl_name = CTL_UNNUMBERED,
> + .procname = "sleep",
> + .data = NULL,
> + .maxlen = 0,
> + .mode = 0600,
> + .proc_handler = &pm_do_sleep
> + },

dito

> + {
> + .ctl_name = CTL_UNNUMBERED,
> + .procname = "freq",
> + .data = NULL,
> + .maxlen = 0,
> + .mode = 0600,
> + .proc_handler = &pm_do_freq
> + },
> + {}
> };

dito

Regards

Ingo Oeser

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

Subject: Re: [PATCH 45/59] sysctl: C99 convert ctl_tables in drivers/parport/procfs.c
Posted by ebiederm on Tue, 16 Jan 2007 23:00:56 GMT
View Forum Message <> Reply to Message

Ingo Oeser <ioe-lkml@rameria.de> writes:

> Hi Eric,
>
> On Tuesday, 16. January 2007 17:39, Eric W. Biederman wrote:
>> diff --git a/drivers/parport/procfs.c b/drivers/parport/procfs.c
>> index 2e744a2..5337789 100644
>> --- a/drivers/parport/procfs.c
>> +++ b/drivers/parport/procfs.c
>> @@ -263,50 +263,118 @@ struct parport_sysctl_table {
>> + {
>> + .ctl_name = DEV_PARPORT_BASE_ADDR,
>> + .procname = "base-addr",
>> + .data = NULL,
>> + .maxlen = 0,
>> + .mode = 0444,
>> + .proc_handler = &do_hardware_base_addr
>> + },
>
> No need to initialize to zero or NULL. Just list any variable, which is NOT zero
> or NULL.

Agreed. In this case it was left for clarity.

>> + {
>> + .ctl_name = DEV_PARPORT_AUTOPROBE + 1,
>> + .procname = "autoprobe0",
>> + .data = NULL,
>> + .maxlen = 0,
>> + .maxlen = 0444,
>> + .proc_handler = &do_autoprobe
>> + },
>
> Typo here? .mode = 0444 makes mor sense.

Yep looks like it. On my todo.

Eric

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

Subject: Re: [PATCH 18/59] sysctl: ipmi remove unnecessary insert_at_head flag
Posted by Benjamin Herrenschmid on Wed, 17 Jan 2007 03:14:54 GMT
View Forum Message <> Reply to Message

On Tue, 2007-01-16 at 09:39 -0700, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

>
> With unique sysctl binary numbers setting insert_at_head is pointless.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Acked-by: Benjamin Herrenschmidt <benh@kernel.crashing.org>

> ---
> drivers/char/ipmi/ipmi_poweroff.c | 2 +-
> 1 files changed, 1 insertions(+), 1 deletions(-)
>
> diff --git a/drivers/char/ipmi/ipmi_poweroff.c b/drivers/char/ipmi/ipmi_poweroff.c
> index 9d23136..b3ae65e 100644
> --- a/drivers/char/ipmi/ipmi_poweroff.c
> +++ b/drivers/char/ipmi/ipmi_poweroff.c
> @@ -686,7 +686,7 @@ static int ipmi_poweroff_init (void)
>   printk(KERN_INFO PFX "Power cycle is enabled.\n");
>
> #ifdef CONFIG_PROC_FS
> - ipmi_table_header = register_sysctl_table(ipmi_root_table, 1);
> + ipmi_table_header = register_sysctl_table(ipmi_root_table, 0);
>   if (!ipmi_table_header) {
>   printk(KERN_ERR PFX "Unable to register powercycle sysctl\n");
>   rv = -ENOMEM;

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

Subject: Re: [PATCH 35/59] sysctl: C99 convert ctl_tables in arch/powerpc/kernel/idle.c

Posted by Benjamin Herrenschmid on Wed, 17 Jan 2007 03:16:27 GMT

View Forum Message <> Reply to Message

On Tue, 2007-01-16 at 09:39 -0700, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> This was partially done already and there was no ABI breakage what
> a relief.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Acked-by: Benjamin Herrenschmidt <benh@kernel.crashing.org>

> ---
> arch/powerpc/kernel/idle.c | 11 ++++++++---

> 1 files changed, 8 insertions(+), 3 deletions(-)
>
> diff --git a/arch/powerpc/kernel/idle.c b/arch/powerpc/kernel/idle.c
> index 8994af3..8b27bb1 100644
> --- a/arch/powerpc/kernel/idle.c
> +++ b/arch/powerpc/kernel/idle.c
> @@ -110,11 +110,16 @@ static ctl_table powersave_nap_ctl_table[]={
>   .mode = 0644,
>   .proc_handler = &proc_dointvec,
>   },
> - { 0, },
> + {}
> };
> static ctl_table powersave_nap_sysctl_root[] = {
> - { 1, "kernel", NULL, 0, 0755, powersave_nap_ctl_table, },
> - { 0,},
> + {
> + .ctl_name = CTL_KERN,
> + .procname = "kernel",
> + .mode = 0755,
> + .child = powersave_nap_ctl_table,
> + },
> + {}
> };
>
> static int __init

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

Subject: Re: [PATCH 36/59] sysctl: C99 convert ctl_tables entries in arch/ppc/kernel/ppc_htab.c
Posted by Benjamin Herrenschmid on Wed, 17 Jan 2007 03:16:54 GMT
View Forum Message <> Reply to Message

On Tue, 2007-01-16 at 09:39 -0700, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> And make the mode of the kernel directory 0555 no one is allowed
> to write to sysctl directories.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Acked-by: Benjamin Herrenschmidt <benh@kernel.crashing.org>

> ---
> arch/ppc/kernel/ppc_htab.c | 11 ++++++++---
> 1 files changed, 8 insertions(+), 3 deletions(-)
>
> diff --git a/arch/ppc/kernel/ppc_htab.c b/arch/ppc/kernel/ppc_htab.c
> index bd129d3..77b20ff 100644
> --- a/arch/ppc/kernel/ppc_htab.c
> +++ b/arch/ppc/kernel/ppc_htab.c
> @@ -442,11 +442,16 @@ static ctl_table htab_ctl_table[]={
>   .mode = 0644,
>   .proc_handler = &proc_dol2crvec,
>   },
> - { 0, },
> + {}
> };
> static ctl_table htab_sysctl_root[] = {
> - { 1, "kernel", NULL, 0, 0755, htab_ctl_table, },
> - { 0,},
> + {
> + .ctl_name = CTL_KERN,
> + .procname = "kernel",
> + .mode = 0555,
> + .child = htab_ctl_table,
> + },
> + {}
> };
>
> static int __init

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by Andi Kleen on Wed, 17 Jan 2007 04:21:43 GMT
View Forum Message <> Reply to Message

On Wednesday 17 January 2007 03:33, Eric W. Biederman wrote:
> There has not been much maintenance on sysctl in years, and as a result is
> there is a lot to do to allow future interesting work to happen, and being
> ambitious I'm trying to do it all at once :)
>
> The patches in this series fall into several general categories.

[...]

The patches look good to me.

-Andi

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by Martin Schwidefsky on Wed, 17 Jan 2007 12:10:13 GMT
View Forum Message <> Reply to Message

On Tue, 2007-01-16 at 09:33 -0700, Eric W. Biederman wrote:
> There has not been much maintenance on sysctl in years, and as a result is
> there is a lot to do to allow future interesting work to happen, and being
> ambitious I'm trying to do it all at once :)

s390 parts look good. Kernels boots and the system controls are still
working. I had to add an #include <linux/uaccess.h> to ipc/ipc_sysctl.c
to get the kernel compiled. That include should be added to patch #51.

Acked-by: Martin Schwidefsky <schwidefsky@de.ibm.com> for:
[PATCH 33/59] sysctl: s390 move sysctl definitions to sysctl.h
[PATCH 34/59] sysctl: s390 Remove unnecessary use of insert_at_head

and the s390 parts of
[PATCH 55/59] sysctl: Remove insert_at_head from register_sysctl

--
blue skies,
  Martin.

Martin Schwidefsky
Linux for zSeries Development & Services
IBM Deutschland Entwicklung GmbH

"Reality continues to ruin my life." - Calvin.

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

Subject: Re: [PATCH 14/59] sysctl: C99 convert xfs ctl_tables

minor extra space in table below...

Kirill

> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> ---
> fs/xfs/linux-2.6/xfs_sysctl.c | 258 ++++++

> - &sysctl_intvec, NULL,
> - &xfs_params.syncd_timer.min, &xfs_params.syncd_timer.max},
> -
> - {XFS_INHERIT_SYNC, "inherit_sync", &xfs_params.inherit_sync.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.inherit_sync.min, &xfs_params.inherit_sync.max},
> -
> - {XFS_INHERIT_NODUMP, "inherit_nodump", &xfs_params.inherit_nodump.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.inherit_nodump.min, &xfs_params.inherit_nodump.max},
> -
> - {XFS_INHERIT_NOATIME, "inherit_noatime", &xfs_params.inherit_noatim.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.inherit_noatim.min, &xfs_params.inherit_noatim.max},
> -
> - {XFS_BUF_TIMER, "xfsbufd_centisecs", &xfs_params.xfs_buf_timer.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.xfs_buf_timer.min, &xfs_params.xfs_buf_timer.max},
> -
> - {XFS_BUF_AGE, "age_buffer_centisecs", &xfs_params.xfs_buf_age.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.xfs_buf_age.min, &xfs_params.xfs_buf_age.max},
> -
> - {XFS_INHERIT_NOSYM, "inherit_nosymlinks", &xfs_params.inherit_nosym.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.inherit_nosym.min, &xfs_params.inherit_nosym.max},
> -
> - {XFS_ROTORSTEP, "rotorstep", &xfs_params.rotorstep.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.rotorstep.min, &xfs_params.rotorstep.max},
> -
> - {XFS_INHERIT_NODFRG, "inherit_nodefrag", &xfs_params.inherit_nodfrg.val,
> - sizeof(int), 0644, NULL, &proc_dointvec_minmax,
> - &sysctl_intvec, NULL,
> - &xfs_params.inherit_nodfrg.min, &xfs_params.inherit_nodfrg.max},
> + {
> + .ctl_name = XFS_RESTRICT_CHOWN,
> + .procname = "restrict_chown",
> + .data = &xfs_params.restrict_chown.val,
> + .maxlen = sizeof(int),
> + .mode = 0644,

```
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.restrict_chown.min,
> +  .extra2  = &xfs_params.restrict_chown.max
> + },
> + {
> +  .ctl_name = XFS_SGID_INHERIT,
> +  .procname = "irix_sgid_inherit",
> +  .data  = &xfs_params.sgid_inherit.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.sgid_inherit.min,
> +  .extra2  = &xfs_params.sgid_inherit.max
> + },
> + {
> +  .ctl_name = XFS_SYMLINK_MODE,
> +  .procname = "irix_symlink_mode",
> +  .data  = &xfs_params.symlink_mode.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.symlink_mode.min,
> +  .extra2  = &xfs_params.symlink_mode.max
> + },
> + {
> +  .ctl_name = XFS_PANIC_MASK,
> +  .procname = "panic_mask",
> +  .data  = &xfs_params.panic_mask.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler =  &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.panic_mask.min,
> +  .extra2  = &xfs_params.panic_mask.max
> + },
>
> + {
> +  .ctl_name = XFS_ERRLEVEL,
> +  .procname = "error_level",
> +  .data  = &xfs_params.error_level.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.error_level.min,
```

```
> +  .extra2  = &xfs_params.error_level.max
> + },
> + {
> +  .ctl_name = XFS_SYNCD_TIMER,
> +  .procname = "xfssyncd_centisecs",
> +  .data  = &xfs_params.syncd_timer.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.syncd_timer.min,
> +  .extra2  = &xfs_params.syncd_timer.max
> + },
> + {
> +  .ctl_name = XFS_INHERIT_SYNC,
> +  .procname = "inherit_sync",
> +  .data  = &xfs_params.inherit_sync.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.inherit_sync.min,
> +  .extra2  = &xfs_params.inherit_sync.max
> + },
> + {
> +  .ctl_name = XFS_INHERIT_NODUMP,
> +  .procname = "inherit_nodump",
> +  .data  = &xfs_params.inherit_nodump.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec, NULL,
> +  .extra1  = &xfs_params.inherit_nodump.min,
> +  .extra2  = &xfs_params.inherit_nodump.max
> + },
> + {
> +  .ctl_name = XFS_INHERIT_NOATIME,
> +  .procname = "inherit_noatime",
> +  .data  = &xfs_params.inherit_noatim.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec, NULL,
> +  .extra1  = &xfs_params.inherit_noatim.min,
> +  .extra2  = &xfs_params.inherit_noatim.max
> + },
> + {
> +  .ctl_name = XFS_BUF_TIMER,
```

```
> +  .procname = "xfsbufd_centisecs",
> +  .data  = &xfs_params.xfs_buf_timer.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.xfs_buf_timer.min,
> +  .extra2  = &xfs_params.xfs_buf_timer.max
> + },
> + {
> +  .ctl_name = XFS_BUF_AGE,
> +  .procname = "age_buffer_centisecs",
> +  .data  = &xfs_params.xfs_buf_age.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec, NULL,
> +  .extra1  = &xfs_params.xfs_buf_age.min,
> +  .extra2  = &xfs_params.xfs_buf_age.max
> + },
> + {
> +  .ctl_name = XFS_INHERIT_NOSYM,
> +  .procname = "inherit_nosymlinks",
> +  .data  = &xfs_params.inherit_nosym.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.inherit_nosym.min,
> +  .extra2  = &xfs_params.inherit_nosym.max
> + },
> + {
> +  .ctl_name = XFS_ROTORSTEP,
> +  .procname = "rotorstep",
> +  .data  = &xfs_params.rotorstep.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.rotorstep.min,
> +  .extra2  = &xfs_params.rotorstep.max
> + },
> + {
> +  .ctl_name = XFS_INHERIT_NODFRG,
> +  .procname = "inherit_nodefrag",
> +  .data  = &xfs_params.inherit_nodfrg.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
```

```
> +  .proc_handler = &proc_dointvec_minmax,
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.inherit_nodfrg.min,
> +  .extra2  = &xfs_params.inherit_nodfrg.max
> + },
>    /* please keep this the last entry */
>  #ifdef CONFIG_PROC_FS
> - {XFS_STATS_CLEAR, "stats_clear", &xfs_params.stats_clear.val,
> - sizeof(int), 0644, NULL, &xfs_stats_clear_proc_handler,
> - &sysctl_intvec, NULL,
> - &xfs_params.stats_clear.min, &xfs_params.stats_clear.max},
> + {
> +  .ctl_name = XFS_STATS_CLEAR,
> +  .procname = "stats_clear",
> +  .data  = &xfs_params.stats_clear.val,
> +  .maxlen  = sizeof(int),
> +  .mode  = 0644,
> +  .proc_handler =  &xfs_stats_clear_proc_handler,
<<< minor. extra space.
> +  .strategy = &sysctl_intvec,
> +  .extra1  = &xfs_params.stats_clear.min,
> +  .extra2  = &xfs_params.stats_clear.max
> + },
>  #endif /* CONFIG_PROC_FS */
>
> - {0}
> + {}
>  };
>
>  STATIC ctl_table xfs_dir_table[] = {
> - {FS_XFS, "xfs", NULL, 0, 0555, xfs_table},
> - {0}
> + {
> +  .ctl_name = FS_XFS,
> +  .procname = "xfs",
> +  .mode  = 0555,
> +  .child  = xfs_table
> + },
> + {}
>  };
>
>  STATIC ctl_table xfs_root_table[] = {
> - {CTL_FS, "fs",  NULL, 0, 0555, xfs_dir_table},
> - {0}
> + {
> +  .ctl_name = CTL_FS,
> +  .procname = "fs",
> +  .mode  = 0555,
```

> + .child = xfs_dir_table
> + },
> + {}
> };
>
> void

_____

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

Subject: Re: [PATCH 25/59] sysctl: C99 convert arch/frv/kernel/pm.c
Posted by dev on Wed, 17 Jan 2007 17:14:17 GMT
View Forum Message <> Reply to Message

another small minor note.

> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> ---
> arch/frv/kernel/pm.c | 50 ++-------
> 1 files changed, 43 insertions(+), 7 deletions(-)
>
> diff --git a/arch/frv/kernel/pm.c b/arch/frv/kernel/pm.c
> index c1840d6..aa50333 100644
> --- a/arch/frv/kernel/pm.c
> +++ b/arch/frv/kernel/pm.c
> @@ -401,17 +401,53 @@ static int cm_sysctl(ctl_table *table, int __user *name, int nlen,
>
> static struct ctl_table pm_table[] =
> {
> - {CTL_PM_SUSPEND, "suspend", NULL, 0, 0200, NULL, &sysctl_pm_do_suspend},
> - {CTL_PM_CMODE, "cmode", &clock_cmode_current, sizeof(int), 0644, NULL,
&cmode_procctl, &cmode_sysctl, NULL},
> - {CTL_PM_P0, "p0", &clock_p0_current, sizeof(int), 0644, NULL, &p0_procctl, &p0_sysctl,
NULL},
> - {CTL_PM_CM, "cm", &clock_cm_current, sizeof(int), 0644, NULL, &cm_procctl, &cm_sysctl,
NULL},
> - {0}
> + {
> + .ctl_name = CTL_PM_SUSPEND,
> + .procname = "suspend",
> + .data = NULL,
> + .maxlen = 0,
> + .mode = 0200,

> + .proc_handler = &sysctl_pm_do_suspend,
> + },
> + {
> + .ctl_name = CTL_PM_CMODE,
> + .procname = "cmode",
> + .data = &clock_cmode_current,
> + .maxlen = sizeof(int),
> + .mode = 0644,
> + .proc_handler = &cmode_procctl,
> + .strategy = &cmode_sysctl,
> + },
> + {
> + .ctl_name = CTL_PM_P0,
> + .procname = "p0",
> + .data = &clock_p0_current,
> + .maxlen = sizeof(int),
> + .mode = 0644,
> + .proc_handler = &p0_procctl,
> + .strategy = &p0_sysctl,
> + },
> + {
> + .ctl_name = CTL_PM_CM,
> + .procname = "cm",
> + .data = &clock_cm_current,
> + .maxlen = sizeof(int),
> + .mode = 0644,
> + .proc_handler = &cm_procctl,
> + .strategy = &cm_sysctl,
> + },
> + { .ctl_name = 0}
in next patch (26/59) you write just "{ }". .ctl_name = 0 not required here.

> };
>
> static struct ctl_table pm_dir_table[] =
> {
> - {CTL_PM, "pm", NULL, 0, 0555, pm_table},
> - {0}
> + {
> + .ctl_name = CTL_PM,
> + .procname = "pm",
> + .mode = 0555,
> + .child = pm_table,
> + },
> + { .ctl_name = 0}
> };
>

> /*

_____

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

Subject: Re: [PATCH 33/59] sysctl: s390 move sysctl definitions to sysctl.h
Posted by dev on Wed, 17 Jan 2007 17:23:22 GMT
View Forum Message <> Reply to Message

IDs not sorted in enum. see below.

> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> We need to have the the definition of all top level sysctl
> directories registers in sysctl.h so we don't conflict by
> accident and cause abi problems.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> ---
> arch/s390/appldata/appldata.h | 3 +--
> arch/s390/kernel/debug.c | 1 -
> arch/s390/mm/cmm.c | 4 ----
> include/linux/sysctl.h | 7 +++++++
> 4 files changed, 8 insertions(+), 7 deletions(-)
>
> diff --git a/arch/s390/appldata/appldata.h b/arch/s390/appldata/appldata.h
> index 0429481..4069b81 100644
> --- a/arch/s390/appldata/appldata.h
> +++ b/arch/s390/appldata/appldata.h
> @@ -21,8 +21,7 @@
> #define APPLDATA_RECORD_NET_SUM_ID 0x03 /* must be < 256 ! */
> #define APPLDATA_RECORD_PROC_ID 0x04
>
> -#define CTL_APPLDATA 2120 /* sysctl IDs, must be unique */
> -#define CTL_APPLDATA_TIMER 2121
> +#define CTL_APPLDATA_TIMER 2121 /* sysctl IDs, must be unique */
> #define CTL_APPLDATA_INTERVAL 2122
> #define CTL_APPLDATA_MEM 2123
> #define CTL_APPLDATA_OS 2124
> diff --git a/arch/s390/kernel/debug.c b/arch/s390/kernel/debug.c
> index bb57bc0..c81f8e5 100644
> --- a/arch/s390/kernel/debug.c
> +++ b/arch/s390/kernel/debug.c
> @@ -852,7 +852,6 @@ debug_finish_entry(debug_info_t * id, debug_entry_t* active, int level,
> static int debug_stoppable=1;

> static int debug_active=1;
>
> -#define CTL_S390DBF 5677
> #define CTL_S390DBF_STOPPABLE 5678
> #define CTL_S390DBF_ACTIVE 5679
>
> diff --git a/arch/s390/mm/cmm.c b/arch/s390/mm/cmm.c
> index 607f50e..df733d5 100644
> --- a/arch/s390/mm/cmm.c
> +++ b/arch/s390/mm/cmm.c
> @@ -256,10 +256,6 @@ cmm_skip_blanks(char *cp, char **endp)
> }
>
> #ifdef CONFIG_CMM_PROC
> -/* These will someday get removed. */
> -#define VM_CMM_PAGES 1111
> -#define VM_CMM_TIMED_PAGES 1112
> -#define VM_CMM_TIMEOUT 1113
>
> static struct ctl_table cmm_table[];
>
> diff --git a/include/linux/sysctl.h b/include/linux/sysctl.h
> index 71c16b4..56d0161 100644
> --- a/include/linux/sysctl.h
> +++ b/include/linux/sysctl.h
> @@ -73,6 +73,8 @@ enum
>   CTL_SUNRPC=7249, /* sunrpc debug */
>   CTL_PM=9899, /* frv power management */
>   CTL_FRV=9898, /* frv specific sysctls */
> + CTL_S390DBF=5677, /* s390 debug */
> + CTL_APPLDATA=2120, /* s390 appldata */
<<<< not sorted by ID? imho should be sorted. otherwise can'be unnotied when inserted above.

> };
>
> /* CTL_BUS names: */
> @@ -205,6 +207,11 @@ enum
>   VM_PANIC_ON_OOM=33, /* panic at out-of-memory */
>   VM_VDSO_ENABLED=34, /* map VDSO into new processes? */
>   VM_MIN_SLAB=35,   /* Percent pages ignored by zone reclaim */
> +
> + /* s390 vm cmm sysctls */
> + VM_CMM_PAGES=1111,
> + VM_CMM_TIMED_PAGES=1112,
> + VM_CMM_TIMEOUT=1113,
> };
>
>

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

Subject: Re: [PATCH 50/59] sysctl: Move utsname sysctls to their own file
Posted by dev on Wed, 17 Jan 2007 17:41:48 GMT
View Forum Message <> Reply to Message

Eric, though I personally don't care much:
1. I ask for not setting your authorship/copyright on the code which you just copied
  from other places. Just doesn't look polite IMHO.
2. I would propose to not introduce utsname_sysctl.c.
  both files are too small and minor that I can't see much reasons splitting them.

Kirill

> 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 utsname logic to together it makes
> the code a little more readable.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> ---
> kernel/Makefile | 1 +
> kernel/sysctl.c | 115 ------------------------------------
> kernel/utsname_sysctl.c | 146 ++++++

> +++ b/kernel/sysctl.c
> @@ -135,13 +135,6 @@ static int parse_table(int __user *, int, void __user *, size_t __user *,
>   void __user *, size_t, ctl_table *);
> #endif
>
> -static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
> -   void __user *buffer, size_t *lenp, loff_t *ppos);
> -
> -static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
> -   void __user *oldval, size_t __user *oldlenp,
> -   void __user *newval, size_t newlen);
> -
> #ifdef CONFIG_SYSVIPC
> static int sysctl_ipc_data(ctl_table *table, int __user *name, int nlen,
>   void __user *oldval, size_t __user *oldlenp,
> @@ -174,27 +167,6 @@ extern ctl_table inotify_table[];
> int sysctl_legacy_va_layout;
> #endif
>
> -static void *get_uts(ctl_table *table, int write)
> -{
> - char *which = table->data;
> -#ifdef CONFIG_UTS_NS
> - struct uts_namespace *uts_ns = current->nsproxy->uts_ns;
> - which = (which - (char *)&init_uts_ns) + (char *)uts_ns;
> -#endif
> - if (!write)
> - down_read(&uts_sem);
> - else
> - down_write(&uts_sem);
> - return which;
> -}
> -
> -static void put_uts(ctl_table *table, int write, void *which)
> -{
> - if (!write)
> - up_read(&uts_sem);
> - else
> - up_write(&uts_sem);
> -}
>
> #ifdef CONFIG_SYSVIPC
> static void *get_ipc(ctl_table *table, int write)
> @@ -275,51 +247,6 @@ static ctl_table root_table[] = {
>
> static ctl_table kern_table[] = {
>   {
> - .ctl_name = KERN_OSTYPE,

> - .procname = "ostype",
> - .data = init_uts_ns.name.sysname,
> - .maxlen = sizeof(init_uts_ns.name.sysname),
> - .mode = 0444,
> - .proc_handler = &proc_do_uts_string,
> - .strategy = &sysctl_uts_string,
> - },
> - {
> - .ctl_name = KERN_OSRELEASE,
> - .procname = "osrelease",
> - .data = init_uts_ns.name.release,
> - .maxlen = sizeof(init_uts_ns.name.release),
> - .mode = 0444,
> - .proc_handler = &proc_do_uts_string,
> - .strategy = &sysctl_uts_string,
> - },
> - {
> - .ctl_name = KERN_VERSION,
> - .procname = "version",
> - .data = init_uts_ns.name.version,
> - .maxlen = sizeof(init_uts_ns.name.version),
> - .mode = 0444,
> - .proc_handler = &proc_do_uts_string,
> - .strategy = &sysctl_uts_string,
> - },
> - {
> - .ctl_name = KERN_NODENAME,
> - .procname = "hostname",
> - .data = init_uts_ns.name.nodename,
> - .maxlen = sizeof(init_uts_ns.name.nodename),
> - .mode = 0644,
> - .proc_handler = &proc_do_uts_string,
> - .strategy = &sysctl_uts_string,
> - },
> - {
> - .ctl_name = KERN_DOMAINNAME,
> - .procname = "domainname",
> - .data = init_uts_ns.name.domainname,
> - .maxlen = sizeof(init_uts_ns.name.domainname),
> - .mode = 0644,
> - .proc_handler = &proc_do_uts_string,
> - .strategy = &sysctl_uts_string,
> - },
> - {
> .ctl_name = KERN_PANIC,
> .procname = "panic",
> .data = &panic_timeout,
> @@ -1746,21 +1673,6 @@ int proc_dostring(ctl_table *table, int write, struct file *filp,

> buffer, lenp, ppos);
> }
>
> -/*
> - * Special case of dostring for the UTS structure. This has locks
> - * to observe. Should this be in kernel/sys.c ????
> - */
> -
> -static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
> -   void __user *buffer, size_t *lenp, loff_t *ppos)
> -{
> - int r;
> - void *which;
> - which = get_uts(table, write);
> - r = _proc_do_string(which, table->maxlen,write,filp,buffer,lenp, ppos);
> - put_uts(table, write, which);
> - return r;
> -}
>
> static int do_proc_dointvec_conv(int *negp, unsigned long *lvalp,
>   int *valp,
> @@ -2379,12 +2291,6 @@ int proc_dostring(ctl_table *table, int write, struct file *filp,
>   return -ENOSYS;
> }
>
> -static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
> - void __user *buffer, size_t *lenp, loff_t *ppos)
> -{
> - 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)
> @@ -2602,21 +2508,6 @@ int sysctl_ms_jiffies(ctl_table *table, int __user *name, int nlen,
> }
>
>
> -/* The generic string strategy routine: */
> -static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
> -   void __user *oldval, size_t __user *oldlenp,
> -   void __user *newval, size_t newlen)
> -{
> - struct ctl_table uts_table;
> - int r, write;
> - write = newval && newlen;
> - memcpy(&uts_table, table, sizeof(uts_table));
> - uts_table.data = get_uts(table, write);

> - r = sysctl_string(&uts_table, name, nlen,
> - oldval, oldlenp, newval, newlen);
> - put_uts(table, write, uts_table.data);
> - return r;
> -}
>
> #ifdef CONFIG_SYSVIPC
> /* The generic sysctl ipc data routine. */
> @@ -2723,12 +2614,6 @@ int sysctl_ms_jiffies(ctl_table *table, int __user *name, int nlen,
>   return -ENOSYS;
> }
>
> -static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
> -   void __user *oldval, size_t __user *oldlenp,
> -   void __user *newval, size_t newlen)
> -{
> - 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)
> diff --git a/kernel/utsname_sysctl.c b/kernel/utsname_sysctl.c
> new file mode 100644
> index 0000000..324aa13
> --- /dev/null
> +++ b/kernel/utsname_sysctl.c
> @@ -0,0 +1,146 @@
> +/*
> + * 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/uts.h>
> +#include <linux/utsname.h>
> +#include <linux/version.h>
> +#include <linux/sysctl.h>
> +
> +static void *get_uts(ctl_table *table, int write)
> +{
> + char *which = table->data;
> +#ifdef CONFIG_UTS_NS

> + struct uts_namespace *uts_ns = current->nsproxy->uts_ns;
> + which = (which - (char *)&init_uts_ns) + (char *)uts_ns;
> +#endif
> + if (!write)
> + down_read(&uts_sem);
> + else
> + down_write(&uts_sem);
> + return which;
> +}
> +
> +static void put_uts(ctl_table *table, int write, void *which)
> +{
> + if (!write)
> + up_read(&uts_sem);
> + else
> + up_write(&uts_sem);
> +}
> +
> +#ifdef CONFIG_PROC_FS
> +/*
> + * Special case of dostring for the UTS structure. This has locks
> + * to observe. Should this be in kernel/sys.c ????
> + */
> +static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
> +   void __user *buffer, size_t *lenp, loff_t *ppos)
> +{
> + struct ctl_table uts_table;
> + int r;
> + memcpy(&uts_table, table, sizeof(uts_table));
> + uts_table.data = get_uts(table, write);
> + r = proc_dostring(&uts_table,write,filp,buffer,lenp, ppos);
> + put_uts(table, write, uts_table.data);
> + return r;
> +}
> +#else
> +#define proc_do_uts_string NULL
> +#endif
> +
> +
> +#ifdef CONFIG_SYSCTL_SYSCALL
> +/* The generic string strategy routine: */
> +static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
> +   void __user *oldval, size_t __user *oldlenp,
> +   void __user *newval, size_t newlen)
> +{
> + struct ctl_table uts_table;
> + int r, write;
> + write = newval && newlen;

```
> + memcpy(&uts_table, table, sizeof(uts_table));
> + uts_table.data = get_uts(table, write);
> + r = sysctl_string(&uts_table, name, nlen,
> +  oldval, oldlenp, newval, newlen);
> + put_uts(table, write, uts_table.data);
> + return r;
> +}
> +#else
> +#define sysctl_uts_string NULL
> +#endif
> +
> +static struct ctl_table uts_kern_table[] = {
> + {
> +  .ctl_name = KERN_OSTYPE,
> +  .procname = "ostype",
> +  .data  = init_uts_ns.name.sysname,
> +  .maxlen  = sizeof(init_uts_ns.name.sysname),
> +  .mode  = 0444,
> +  .proc_handler = proc_do_uts_string,
> +  .strategy = sysctl_uts_string,
> + },
> + {
> +  .ctl_name = KERN_OSRELEASE,
> +  .procname = "osrelease",
> +  .data  = init_uts_ns.name.release,
> +  .maxlen  = sizeof(init_uts_ns.name.release),
> +  .mode  = 0444,
> +  .proc_handler = proc_do_uts_string,
> +  .strategy = sysctl_uts_string,
> + },
> + {
> +  .ctl_name = KERN_VERSION,
> +  .procname = "version",
> +  .data  = init_uts_ns.name.version,
> +  .maxlen  = sizeof(init_uts_ns.name.version),
> +  .mode  = 0444,
> +  .proc_handler = proc_do_uts_string,
> +  .strategy = sysctl_uts_string,
> + },
> + {
> +  .ctl_name = KERN_NODENAME,
> +  .procname = "hostname",
> +  .data  = init_uts_ns.name.nodename,
> +  .maxlen  = sizeof(init_uts_ns.name.nodename),
> +  .mode  = 0644,
> +  .proc_handler = proc_do_uts_string,
> +  .strategy = sysctl_uts_string,
> + },
```

```
> + {
> +  .ctl_name = KERN_DOMAINNAME,
> +  .procname = "domainname",
> +  .data  = init_uts_ns.name.domainname,
> +  .maxlen  = sizeof(init_uts_ns.name.domainname),
> +  .mode  = 0644,
> +  .proc_handler = proc_do_uts_string,
> +  .strategy = sysctl_uts_string,
> + },
> + {}
> +};
> +
> +static struct ctl_table uts_root_table[] = {
> + {
> +  .ctl_name = CTL_KERN,
> +  .procname = "kernel",
> +  .mode  = 0555,
> +  .child  = uts_kern_table,
> + },
> + {}
> +};
> +
> +static int __init utsname_sysctl_init(void)
> +{
> + register_sysctl_table(uts_root_table, 0);
> + return 0;
> +}
> +
> +__initcall(utsname_sysctl_init);
```

_____

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

Subject: Re: [PATCH 51/59] sysctl: Move SYSV IPC sysctls to their own file
Posted by dev on Wed, 17 Jan 2007 17:44:34 GMT
View Forum Message <> Reply to Message

1. I ask for not setting your authorship/copyright on the code which you just copied
   from other places. Just doesn't look polite IMHO.
2. please don't name files like ipc/ipc_sysctl.c
   ipc/sysctl.c sounds better IMHO.
3. any reason to introduce CONFIG_SYSVIPC_SYSCTL?
   why not simply do
   > +obj-$(CONFIG_SYSCTL) += sysctl.o
   instead?

Kirill

> 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 */
>
>  /*
```

_____

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by dev on Wed, 17 Jan 2007 18:10:50 GMT

Eric, really good job!

Patches: 1-13, 15-24, 26-32, 34-44, 46-49, 52-55, 57 (all except below)
Acked-By: Kirill Korotaev <dev@openvz.org>

14/59 - minor (extra space)
25/59 - minor note
33/59 - not sorted sysctl IDs
45/59 - typo
50/59 - copyright/file note
51/59 - copyright/file name/kconfig option notes

56,58,59/59 - will review tomorrow

another issue I have to think over is removal of de->owner.
Alexey Dobriyan has sent recently patching fixing /proc <-> modules refcounting.
I guess w/o these patches your changes are not safe if proc_handler or strategy
are functions from the module.

Thanks,
Kirill

> There has not been much maintenance on sysctl in years, and as a result is
> there is a lot to do to allow future interesting work to happen, and being
> ambitious I'm trying to do it all at once :)
>
> The patches in this series fall into several general categories.
>
> - Removal of useless attempts to override the standard sysctls
>
> - Registers of sysctl numbers in sysctl.h so someone else does not use
>   the magic number and conflict.
>
> - C99 conversions so it becomes possible to change the layout of
>   struct ctl_table without breaking everything.
>
> - Removal of useless claims of module ownership, in the proc dir entries
>
> - Removal of sys_sysctl support where people had used conflicting sysctl
>   numbers. Trying to break glibc or other applications by changing the
>   ABI is not cool. 9 instances of this in the kernel seems a little
>   extreme.
>
> - General enhancements when I got the junk I could see out.
>
> Odds are I missed something, most of the cleanups are simply a result of
> me working on the sysctl core and glancing at the users and going: What?

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

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

Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by ebiederm on Wed, 17 Jan 2007 19:02:27 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> Eric, really good job!
>
> Patches: 1-13, 15-24, 26-32, 34-44, 46-49, 52-55, 57 (all except below)
> Acked-By: Kirill Korotaev <dev@openvz.org>
>
> 14/59 - minor (extra space)
> 25/59 - minor note
> 33/59 - not sorted sysctl IDs
> 45/59 - typo
> 50/59 - copyright/file note
> 51/59 - copyright/file name/kconfig option notes
>
> 56,58,59/59 - will review tomorrow
>
> another issue I have to think over is removal of de->owner.
> Alexey Dobriyan has sent recently patching fixing /proc <-> modules refcounting.
> I guess w/o these patches your changes are not safe if proc_handler or strategy
> are functions from the module.

sysctl uses the logic in use_table/unuse_table to keep it safe from module
remove while it is in use. And it does the logic in the generic code
in either do_rw_proc or do_sysctl. This definitely works on the sys_sysctl path
and it appears to work in the do_rw_proc case, things are a little trickier
there so someone may have missed a race somewhere. In my rewrite of proc
it works exactly like the binary case so we are good there.

It is certainly the intention of the sysctl implementation that users

should not have to set de->owner. So if there is a problem with
removing de->owner it is a bug in the sysctl implementation not in
the code where it was removed.

Normal proc users definitely have to set de->owner to be safe, but sysctl has
always been it's own thing, with different rules.

Eric

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

Subject: Re: [PATCH 50/59] sysctl: Move utsname sysctls to their own file
Posted by ebiederm on Wed, 17 Jan 2007 19:31:22 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> Eric, though I personally don't care much:
> 1. I ask for not setting your authorship/copyright on the code which you just
> copied
>   from other places. Just doesn't look polite IMHO.

I can't claim complete ownership of the code, there was plenty of feed back
and contributions from others but the final form without a big switch
statement is mine. I certainly can't claim the table, it has been in
that form for years.

If you notice I actually didn't say whose copyright it was :) just
that I wrote the file.

If there are copyright claims I should include I will be happy to do that.
Mostly I was just trying to find some stupid boiler plate that would work.

> 2. I would propose to not introduce utsname_sysctl.c.
>   both files are too small and minor that I can't see much reasons splitting
> them.

The impact of moving this code out of sysctl.c is a major
simplification, to sysctl.c. Putting them in their own file means we
can cleanly restrict the code to only be compiled CONFIG_SYSCTL is set.

It is a necessary first step to implementing a per process /proc/sys.

It reorganizes the ipc and utsname sysctl from a terribly fragile
structure to something that is robust and easy to follow. Code

scattered all throughout sysctl.c was just a disaster. We had
several instances of having to fix bugs with odd combinations of
CONFIG options, simply because the other spot that needed to be touched
wasn't obvious.

So from my perspective this is an extremely worthwhile change that
will make maintenance easier and is a small first step towards
some nice future functionality.

Eric

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

Subject: Re: [PATCH 25/59] sysctl: C99 convert arch/frv/kernel/pm.c
Posted by Herbert Poetzl on Mon, 22 Jan 2007 22:21:15 GMT
View Forum Message <> Reply to Message

On Wed, Jan 17, 2007 at 08:14:17PM +0300, Kirill Korotaev wrote:
> another small minor note.
>
> > From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
> >
> > Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> > ---
> > arch/frv/kernel/pm.c | 50 ++-------
> > 1 files changed, 43 insertions(+), 7 deletions(-)
> >
> > diff --git a/arch/frv/kernel/pm.c b/arch/frv/kernel/pm.c
> > index c1840d6..aa50333 100644
> > --- a/arch/frv/kernel/pm.c
> > +++ b/arch/frv/kernel/pm.c
> > @@ -401,17 +401,53 @@ static int cm_sysctl(ctl_table *table, int __user *name, int nlen,
> >
> > static struct ctl_table pm_table[] =
> > {
> > - {CTL_PM_SUSPEND, "suspend", NULL, 0, 0200, NULL, &sysctl_pm_do_suspend},
> > - {CTL_PM_CMODE, "cmode", &clock_cmode_current, sizeof(int), 0644, NULL,
&cmode_procctl, &cmode_sysctl, NULL},
> > - {CTL_PM_P0, "p0", &clock_p0_current, sizeof(int), 0644, NULL, &p0_procctl, &p0_sysctl,
NULL},
> > - {CTL_PM_CM, "cm", &clock_cm_current, sizeof(int), 0644, NULL, &cm_procctl, &cm_sysctl,
NULL},
> > - {0}
> > + {
> > + .ctl_name = CTL_PM_SUSPEND,

> > + .procname = "suspend",
> > + .data = NULL,
> > + .maxlen = 0,
> > + .mode = 0200,
> > + .proc_handler = &sysctl_pm_do_suspend,
> > + },
> > + {
> > + .ctl_name = CTL_PM_CMODE,
> > + .procname = "cmode",
> > + .data = &clock_cmode_current,
> > + .maxlen = sizeof(int),
> > + .mode = 0644,
> > + .proc_handler = &cmode_procctl,
> > + .strategy = &cmode_sysctl,
> > + },
> > + {
> > + .ctl_name = CTL_PM_P0,
> > + .procname = "p0",
> > + .data = &clock_p0_current,
> > + .maxlen = sizeof(int),
> > + .mode = 0644,
> > + .proc_handler = &p0_procctl,
> > + .strategy = &p0_sysctl,
> > + },
> > + {
> > + .ctl_name = CTL_PM_CM,
> > + .procname = "cm",
> > + .data = &clock_cm_current,
> > + .maxlen = sizeof(int),
> > + .mode = 0644,
> > + .proc_handler = &cm_procctl,
> > + .strategy = &cm_sysctl,
> > + },
> > + { .ctl_name = 0}
> in next patch (26/59) you write just "{ }". .ctl_name = 0 not required here.

I'd prefer '{ 0 }' here, but I'm fine with the '{ .ctl_name = 0 }'
too, just '{ }' seems confusing, and it actually might get
misinterpreted too ..

best,
Herbert

> > };
> >
> > static struct ctl_table pm_dir_table[] =
> > {
> > - {CTL_PM, "pm", NULL, 0, 0555, pm_table},

> > - {0}
> > + {
> > + .ctl_name = CTL_PM,
> > + .procname = "pm",
> > + .mode = 0555,
> > + .child = pm_table,
> > + },
> > + { .ctl_name = 0}
> > };
> >
> > /*
>
> _____
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers

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

Subject: Re: [PATCH 50/59] sysctl: Move utsname sysctls to their own file
Posted by Herbert Poetzl on Mon, 22 Jan 2007 22:24:42 GMT
View Forum Message <> Reply to Message

On Wed, Jan 17, 2007 at 12:31:22PM -0700, Eric W. Biederman wrote:
> Kirill Korotaev <dev@sw.ru> writes:
>
> > Eric, though I personally don't care much:
> > 1. I ask for not setting your authorship/copyright on the code which you just
> > copied
> > from other places. Just doesn't look polite IMHO.
>
> I can't claim complete ownership of the code, there was plenty of feed back
> and contributions from others but the final form without a big switch
> statement is mine. I certainly can't claim the table, it has been in
> that form for years.
>
> If you notice I actually didn't say whose copyright it was :) just
> that I wrote the file.
>
> If there are copyright claims I should include I will be happy to do that.
> Mostly I was just trying to find some stupid boiler plate that would work.

IMHO that is fine ...

> > 2. I would propose to not introduce utsname_sysctl.c.

> > both files are too small and minor that I can't see much reasons splitting
> > them.
>
> The impact of moving this code out of sysctl.c is a major
> simplification, to sysctl.c. Putting them in their own file means we
> can cleanly restrict the code to only be compiled CONFIG_SYSCTL is set.
>
> It is a necessary first step to implementing a per process /proc/sys.
>
> It reorganizes the ipc and utsname sysctl from a terribly fragile
> structure to something that is robust and easy to follow. Code
> scattered all throughout sysctl.c was just a disaster. We had
> several instances of having to fix bugs with odd combinations of
> CONFIG options, simply because the other spot that needed to be touched
> wasn't obvious.
>
> So from my perspective this is an extremely worthwhile change that
> will make maintenance easier and is a small first step towards
> some nice future functionality.

yep, agreed ...

best,
Herbert

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

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

Subject: Re: [PATCH 25/59] sysctl: C99 convert arch/frv/kernel/pm.c
Posted by David Howells on Wed, 24 Jan 2007 09:00:16 GMT
View Forum Message <> Reply to Message

Fine by me.

Acked-By: David Howells <dhowells@redhat.com>

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

Subject: Re: [PATCH 49/59] sysctl: Move init_irq_proc into init/main where it belongs
Posted by Andrew Morton on Sat, 27 Jan 2007 10:51:37 GMT
View Forum Message <> Reply to Message

On Tue, 16 Jan 2007 09:39:54 -0700
"Eric W. Biederman" <ebiederm@xmission.com> wrote:

> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> ---
> init/main.c | 3 +++
> kernel/sysctl.c | 3 ---
> 2 files changed, 3 insertions(+), 3 deletions(-)
>
> diff --git a/init/main.c b/init/main.c
> index 8b4a7d7..8af5c6e 100644
> --- a/init/main.c
> +++ b/init/main.c
> @@ -691,6 +691,9 @@ static void __init do_basic_setup(void)
> #ifdef CONFIG_SYSCTL
>   sysctl_init();
> #endif
> +#ifdef CONFIG_PROC_FS
> + init_irq_proc();
> +#endif
>
>   do_initcalls();
> }
> diff --git a/kernel/sysctl.c b/kernel/sysctl.c
> index 600b333..7420761 100644
> --- a/kernel/sysctl.c
> +++ b/kernel/sysctl.c
> @@ -1172,8 +1172,6 @@ static ctl_table dev_table[] = {
>   { .ctl_name = 0 }
> };
>
> -extern void init_irq_proc (void);
> -
> static DEFINE_SPINLOCK(sysctl_lock);
>
> /* called under sysctl_lock */
> @@ -1219,7 +1217,6 @@ void __init sysctl_init(void)
> {
> #ifdef CONFIG_PROC_SYSCTL
>   register_proc_table(root_table, proc_sys_root, &root_table_header);
> - init_irq_proc();
> #endif

> }

sparc64:

init/main.c: In function `do_basic_setup':
init/main.c:707: warning: implicit declaration of function `init_irq_proc'

I couldn't be bothered working out how init/main.c is supposed to get at its declaration of init_irq_proc(). It's not allowed to include linux/irq.h.

_____

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