
Subject: Re: [PATCH -mm] uts namespace : remove CONFIG_UTS_NS

Posted by [serue](#) on Tue, 16 Jan 2007 15:27:34 GMT

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

Quoting Cedric Le Goater (clg@fr.ibm.com):

> CONFIG_UTS_NS has very little value as it only deactivates the unshare
> of the uts namespace and does not improve performance.

>

> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

It really is worthless complication, with the sole effect being to
complicate testing...

Acked-by: Serge Hallyn <serue@us.ibm.com>

> ---

> include/linux/utsname.h | 19 -----

> init/Kconfig | 8 -----

> kernel/Makefile | 3 +--

> kernel/sysctl.c | 3 +--

> 4 files changed, 2 insertions(+), 31 deletions(-)

>

> Index: 2.6.20-rc4-mm1/include/linux/utsname.h

> =====

> --- 2.6.20-rc4-mm1.orig/include/linux/utsname.h

> +++ 2.6.20-rc4-mm1/include/linux/utsname.h

> @@ -48,7 +48,6 @@ static inline void get_uts_ns(struct uts

> kref_get(&ns->kref);

> }

>

> -#ifdef CONFIG_UTS_NS

> extern int unshare_utsname(unsigned long unshare_flags,

> struct uts_namespace **new_uts);

> extern int copy_utsname(int flags, struct task_struct *tsk);

> @@ -58,24 +57,6 @@ static inline void put_uts_ns(struct uts

> {

> kref_put(&ns->kref, free_uts_ns);

> }

> -#else

> -static inline int unshare_utsname(unsigned long unshare_flags,

> - struct uts_namespace **new_uts)

> -{

> - if (unshare_flags & CLONE_NEWUTS)

> - return -EINVAL;

> -

> - return 0;

> -}

> -

```

> -static inline int copy_utsname(int flags, struct task_struct *tsk)
> -{
> - return 0;
> -}
> -static inline void put_uts_ns(struct uts_namespace *ns)
> -{
> -}
> -#endif
>
> static inline struct new_utsname *utsname(void)
> {
> Index: 2.6.20-rc4-mm1/init/Kconfig
> =====
> --- 2.6.20-rc4-mm1.orig/init/Kconfig
> +++ 2.6.20-rc4-mm1/init/Kconfig
> @@ -205,14 +205,6 @@ config TASK_DELAY_ACCT
>
> Say N if unsure.
>
> -config UTS_NS
> - bool "UTS Namespaces"
> - default n
> - help
> - Support uts namespaces. This allows containers, i.e.
> - vservers, to use uts namespaces to provide different
> - uts info for different servers. If unsure, say N.
> -
> config AUDIT
> bool "Auditing support"
> depends on NET
> Index: 2.6.20-rc4-mm1/kernel/Makefile
> =====
> --- 2.6.20-rc4-mm1.orig/kernel/Makefile
> +++ 2.6.20-rc4-mm1/kernel/Makefile
> @@ -8,7 +8,7 @@ obj-y = sched.o fork.o exec_domain.o
> signal.o sys.o kmod.o workqueue.o pid.o \
> rcupdate.o extable.o params.o posix-timers.o \
> kthread.o wait.o kfifo.o sys_ni.o posix-cpu-timers.o mutex.o \
> - hrtimer.o rwsem.o latency.o nsproxy.o srcu.o
> + hrtimer.o rwsem.o latency.o nsproxy.o srcu.o utsname.o
>
> obj-$(CONFIG_STACKTRACE) += stacktrace.o
> obj-y += time/
> @@ -48,7 +48,6 @@ obj-$(CONFIG_SECCOMP) += seccomp.o
> obj-$(CONFIG_RCU_TORTURE_TEST) += rcutorture.o
> obj-$(CONFIG_DEBUG_SYNCHRO_TEST) += synchro-test.o
> obj-$(CONFIG_RELAY) += relay.o
> -obj-$(CONFIG_UTS_NS) += utsname.o

```

```
> obj-$(CONFIG_TASK_DELAY_ACCT) += delayacct.o
> obj-$(CONFIG_TASKSTATS) += taskstats.o tsacct.o
>
> Index: 2.6.20-rc4-mm1/kernel/sysctl.c
> =====
> --- 2.6.20-rc4-mm1.orig/kernel/sysctl.c
> +++ 2.6.20-rc4-mm1/kernel/sysctl.c
> @@ -187,10 +187,9 @@ int sysctl_legacy_va_layout;
> 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
```

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

Subject: Re: [PATCH -mm] uts namespace : remove CONFIG_UTS_NS
Posted by [Herbert Poetzl](#) on Mon, 22 Jan 2007 22:09:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, Jan 16, 2007 at 09:27:34AM -0600, Serge E. Hallyn wrote:
> Quoting Cedric Le Goater (clg@fr.ibm.com):
> > CONFIG_UTS_NS has very little value as it only deactivates the unshare
> > of the uts namespace and does not improve performance.
> >
> > Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
>
> It really is worthless complication, with the sole effect being to
> complicate testing...

we already force that on by default (in Linux-VServer)
and I don't think the code can be considered critical
enough for special casing that ...

so I'm perfectly fine with removing that for UTS

best,
Herbert

```

> Acked-by: Serge Hallyn <serue@us.ibm.com>
>
>> ---
>> include/linux/utsname.h | 19 -----
>> init/Kconfig           | 8 -----
>> kernel/Makefile       | 3 +--
>> kernel/sysctl.c       | 3 +--
>> 4 files changed, 2 insertions(+), 31 deletions(-)
>>
>> Index: 2.6.20-rc4-mm1/include/linux/utsname.h
>> =====
>> --- 2.6.20-rc4-mm1.orig/include/linux/utsname.h
>> +++ 2.6.20-rc4-mm1/include/linux/utsname.h
>> @@ -48,7 +48,6 @@ static inline void get_uts_ns(struct uts
>>  kref_get(&ns->kref);
>> }
>>
>> -#ifdef CONFIG_UTS_NS
>> extern int unshare_utsname(unsigned long unshare_flags,
>>  struct uts_namespace **new_uts);
>> extern int copy_utsname(int flags, struct task_struct *tsk);
>> @@ -58,24 +57,6 @@ static inline void put_uts_ns(struct uts
>> {
>>  kref_put(&ns->kref, free_uts_ns);
>> }
>> -#else
>> -static inline int unshare_utsname(unsigned long unshare_flags,
>> - struct uts_namespace **new_uts)
>> -{
>> - if (unshare_flags & CLONE_NEWUTS)
>> - return -EINVAL;
>> -
>> - return 0;
>> -}
>> -
>> -static inline int copy_utsname(int flags, struct task_struct *tsk)
>> -{
>> - return 0;
>> -}
>> -static inline void put_uts_ns(struct uts_namespace *ns)
>> -{
>> -}
>> -#endif
>>
>> static inline struct new_utsname *utsname(void)
>> {
>> Index: 2.6.20-rc4-mm1/init/Kconfig
>> =====

```

```

>> --- 2.6.20-rc4-mm1.orig/init/Kconfig
>> +++ 2.6.20-rc4-mm1/init/Kconfig
>> @@ -205,14 +205,6 @@ config TASK_DELAY_ACCT
>>
>> Say N if unsure.
>>
>> -config UTS_NS
>> - bool "UTS Namespaces"
>> - default n
>> - help
>> - Support uts namespaces. This allows containers, i.e.
>> - vservers, to use uts namespaces to provide different
>> - uts info for different servers. If unsure, say N.
>> -
>> config AUDIT
>> bool "Auditing support"
>> depends on NET
>> Index: 2.6.20-rc4-mm1/kernel/Makefile
>> =====
>> --- 2.6.20-rc4-mm1.orig/kernel/Makefile
>> +++ 2.6.20-rc4-mm1/kernel/Makefile
>> @@ -8,7 +8,7 @@ obj-y    = sched.o fork.o exec_domain.o
>>    signal.o sys.o kmod.o workqueue.o pid.o \
>>    rcupdate.o extable.o params.o posix-timers.o \
>>    kthread.o wait.o kfifo.o sys_ni.o posix-cpu-timers.o mutex.o \
>> -   hrtimer.o rwsem.o latency.o nsproxy.o srcu.o
>> +   hrtimer.o rwsem.o latency.o nsproxy.o srcu.o utsname.o
>>
>> obj-$(CONFIG_STACKTRACE) += stacktrace.o
>> obj-y += time/
>> @@ -48,7 +48,6 @@ obj-$(CONFIG_SECCOMP) += seccomp.o
>> obj-$(CONFIG_RCU_TORTURE_TEST) += rcutorture.o
>> obj-$(CONFIG_DEBUG_SYNCHRO_TEST) += synchro-test.o
>> obj-$(CONFIG_RELAY) += relay.o
>> -obj-$(CONFIG_UTS_NS) += utsname.o
>> obj-$(CONFIG_TASK_DELAY_ACCT) += delayacct.o
>> obj-$(CONFIG_TASKSTATS) += taskstats.o tsacct.o
>>
>> Index: 2.6.20-rc4-mm1/kernel/sysctl.c
>> =====
>> --- 2.6.20-rc4-mm1.orig/kernel/sysctl.c
>> +++ 2.6.20-rc4-mm1/kernel/sysctl.c
>> @@ -187,10 +187,9 @@ int sysctl_legacy_va_layout;
>> 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
> > _____
```

```
> 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
```
