Subject: Re: [PATCH -mm] ipc namespace : remove CONFIG_IPC_NS
Posted by serue on Tue, 16 Jan 2007 15:17:49 GMT
View Forum Message <> Reply to Message

Quoting Cedric Le Goater (clg@fr.ibm.com):
> CONFIG_IPC_NS has very little value as it only deactivates the unshare
> of the ipc namespace and does not improve performance.
>
> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

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

> ---
> include/linux/ipc.h | 11 -----------
> init/Kconfig | 9 ---------
> ipc/msg.c | 4 +---
> ipc/sem.c | 4 +---
> ipc/shm.c | 4 +---
> ipc/util.c | 4 +---
> ipc/util.h | 8 ++------
> kernel/fork.c | 10 ----------
> 8 files changed, 6 insertions(+), 48 deletions(-)
>
> Index: 2.6.20-rc4-mm1/include/linux/ipc.h
> ===
> --- 2.6.20-rc4-mm1.orig/include/linux/ipc.h
> +++ 2.6.20-rc4-mm1/include/linux/ipc.h
> @@ -96,31 +96,20 @@ extern struct ipc_namespace init_ipc_ns;
> #define INIT_IPC_NS(ns)
> #endif
>
> -#ifdef CONFIG_IPC_NS
> extern void free_ipc_ns(struct kref *kref);
> extern int copy_ipcs(unsigned long flags, struct task_struct *tsk);
> extern int unshare_ipcs(unsigned long flags, struct ipc_namespace **ns);
> -#else
> -static inline int copy_ipcs(unsigned long flags, struct task_struct *tsk)
> -{
> - return 0;
> -}
> -#endif
>
> static inline struct ipc_namespace *get_ipc_ns(struct ipc_namespace *ns)
> {
> -#ifdef CONFIG_IPC_NS
> if (ns)
> kref_get(&ns->kref);
> -#endif

> return ns;
> }
>
> static inline void put_ipc_ns(struct ipc_namespace *ns)
> {
> -#ifdef CONFIG_IPC_NS
> kref_put(&ns->kref, free_ipc_ns);
> -#endif
> }
>
> #endif /* __KERNEL__ */
> Index: 2.6.20-rc4-mm1/init/Kconfig
> ===
> --- 2.6.20-rc4-mm1.orig/init/Kconfig
> +++ 2.6.20-rc4-mm1/init/Kconfig
> @@ -138,15 +138,6 @@ config SYSVIPC
>   section 6.4 of the Linux Programmer's Guide, available from
>   <http://www.tldp.org/guides.html>.
>
> -config IPC_NS
> - bool "IPC Namespaces"
> - depends on SYSVIPC
> - default n
> - help
> - Support ipc namespaces. This allows containers, i.e. virtual
> - environments, to use ipc namespaces to provide different ipc
> - objects for different servers. If unsure, say N.
> -
> config POSIX_MQUEUE
> bool "POSIX Message Queues"
> depends on NET && EXPERIMENTAL
> Index: 2.6.20-rc4-mm1/ipc/msg.c
> ===
> --- 2.6.20-rc4-mm1.orig/ipc/msg.c
> +++ 2.6.20-rc4-mm1/ipc/msg.c
> @@ -87,7 +87,7 @@ static int newque (struct ipc_namespace
> static int sysvipc_msg_proc_show(struct seq_file *s, void *it);
> #endif
>
> -static void __ipc_init __msg_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> +static void __msg_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> {
> ns->ids[IPC_MSG_IDS] = ids;
> ns->msg_ctlmax = MSGMAX;
> @@ -96,7 +96,6 @@ static void __ipc_init __msg_init_ns(str
> ipc_init_ids(ids, ns->msg_ctlmni);
> }
>

```
> -#ifdef CONFIG_IPC_NS
>  int msg_init_ns(struct ipc_namespace *ns)
>  {
>   struct ipc_ids *ids;
> @@ -128,7 +127,6 @@ void msg_exit_ns(struct ipc_namespace *n
>   kfree(ns->ids[IPC_MSG_IDS]);
>   ns->ids[IPC_MSG_IDS] = NULL;
>  }
> -#endif
>
>  void __init msg_init(void)
>  {
> Index: 2.6.20-rc4-mm1/ipc/sem.c
> ===================================================================
> --- 2.6.20-rc4-mm1.orig/ipc/sem.c
> +++ 2.6.20-rc4-mm1/ipc/sem.c
> @@ -122,7 +122,7 @@ static int sysvipc_sem_proc_show(struct
>  #define sc_semopm sem_ctls[2]
>  #define sc_semmni sem_ctls[3]
>
> -static void __ipc_init __sem_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> +static void __sem_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
>  {
>   ns->ids[IPC_SEM_IDS] = ids;
>   ns->sc_semmsl = SEMMSL;
> @@ -133,7 +133,6 @@ static void __ipc_init __sem_init_ns(str
>   ipc_init_ids(ids, ns->sc_semmni);
>  }
>
> -#ifdef CONFIG_IPC_NS
>  int sem_init_ns(struct ipc_namespace *ns)
>  {
>   struct ipc_ids *ids;
> @@ -165,7 +164,6 @@ void sem_exit_ns(struct ipc_namespace *n
>   kfree(ns->ids[IPC_SEM_IDS]);
>   ns->ids[IPC_SEM_IDS] = NULL;
>  }
> -#endif
>
>  void __init sem_init (void)
>  {
> Index: 2.6.20-rc4-mm1/ipc/shm.c
> ===================================================================
> --- 2.6.20-rc4-mm1.orig/ipc/shm.c
> +++ 2.6.20-rc4-mm1/ipc/shm.c
> @@ -67,7 +67,7 @@ static void shm_destroy (struct ipc_name
>  static int sysvipc_shm_proc_show(struct seq_file *s, void *it);
>  #endif
```

>
> -static void __ipc_init __shm_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> +static void __shm_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> {
>   ns->ids[IPC_SHM_IDS] = ids;
>   ns->shm_ctlmax = SHMMAX;
> @@ -88,7 +88,6 @@ static void do_shm_rmid(struct ipc_names
>   shm_destroy(ns, shp);
> }
>
> -#ifdef CONFIG_IPC_NS
> int shm_init_ns(struct ipc_namespace *ns)
> {
>   struct ipc_ids *ids;
> @@ -120,7 +119,6 @@ void shm_exit_ns(struct ipc_namespace *n
>   kfree(ns->ids[IPC_SHM_IDS]);
>   ns->ids[IPC_SHM_IDS] = NULL;
> }
> -#endif
>
> void __init shm_init (void)
> {
> Index: 2.6.20-rc4-mm1/ipc/util.c
> ===
> --- 2.6.20-rc4-mm1.orig/ipc/util.c
> +++ 2.6.20-rc4-mm1/ipc/util.c
> @@ -51,7 +51,6 @@ struct ipc_namespace init_ipc_ns = {
>   },
> };
>
> -#ifdef CONFIG_IPC_NS
> static struct ipc_namespace *clone_ipc_ns(struct ipc_namespace *old_ns)
> {
>   int err;
> @@ -144,7 +143,6 @@ void free_ipc_ns(struct kref *kref)
>   shm_exit_ns(ns);
>   kfree(ns);
> }
> -#endif
>
> /**
>   * ipc_init - initialise IPC subsystem
> @@ -172,7 +170,7 @@ __initcall(ipc_init);
>   * array itself.
>   */
>
> -void __ipc_init ipc_init_ids(struct ipc_ids* ids, int size)
> +void ipc_init_ids(struct ipc_ids* ids, int size)

> {
> int i;
>
> Index: 2.6.20-rc4-mm1/ipc/util.h
> ===
> --- 2.6.20-rc4-mm1.orig/ipc/util.h
> +++ 2.6.20-rc4-mm1/ipc/util.h
> @@ -41,12 +41,8 @@ struct ipc_ids {
> };
>
> struct seq_file;
> -#ifdef CONFIG_IPC_NS
> -#define __ipc_init
> -#else
> -#define __ipc_init __init
> -#endif
> -void __ipc_init ipc_init_ids(struct ipc_ids *ids, int size);
> +
> +void ipc_init_ids(struct ipc_ids *ids, int size);
> #ifdef CONFIG_PROC_FS
> void __init ipc_init_proc_interface(const char *path, const char *header,
> int ids, int (*show)(struct seq_file *, void *));
> Index: 2.6.20-rc4-mm1/kernel/fork.c
> ===
> --- 2.6.20-rc4-mm1.orig/kernel/fork.c
> +++ 2.6.20-rc4-mm1/kernel/fork.c
> @@ -1595,16 +1595,6 @@ static int unshare_semundo(unsigned long
> return 0;
> }
>
> -#ifndef CONFIG_IPC_NS
> -static inline int unshare_ipcs(unsigned long flags, struct ipc_namespace **ns)
> -{
> - if (flags & CLONE_NEWIPC)
> - return -EINVAL;
> -
> - return 0;
> -}
> -#endif
> -
> /*
> * unshare allows a process to 'unshare' part of the process
> * context which was originally shared using clone. copy_*
> _____
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers
_____

Subject: Re: [PATCH -mm] ipc namespace : remove CONFIG_IPC_NS
Posted by Herbert Poetzl on Mon, 22 Jan 2007 22:17:14 GMT
View Forum Message <> Reply to Message

On Tue, Jan 16, 2007 at 09:17:49AM -0600, Serge E. Hallyn wrote:
> Quoting Cedric Le Goater (clg@fr.ibm.com):
> > CONFIG_IPC_NS has very little value as it only deactivates the unshare
> > of the ipc namespace and does not improve performance.
> >
> > Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

same a for the UTS namespace, so I'm fine with
this removal too

best,
Herbert

> Acked-by: Serge Hallyn <serue@us.ibm.com>
>
> > ---
> > include/linux/ipc.h | 11 -----------
> > init/Kconfig | 9 ---------
> > ipc/msg.c | 4 +---
> > ipc/sem.c | 4 +---
> > ipc/shm.c | 4 +---
> > ipc/util.c | 4 +---
> > ipc/util.h | 8 ++------
> > kernel/fork.c | 10 ----------
> > 8 files changed, 6 insertions(+), 48 deletions(-)
> >
> > Index: 2.6.20-rc4-mm1/include/linux/ipc.h
> > ===
> > --- 2.6.20-rc4-mm1.orig/include/linux/ipc.h
> > +++ 2.6.20-rc4-mm1/include/linux/ipc.h
> > @@ -96,31 +96,20 @@ extern struct ipc_namespace init_ipc_ns;
> > #define INIT_IPC_NS(ns)
> > #endif
> >
> > -#ifdef CONFIG_IPC_NS
> > extern void free_ipc_ns(struct kref *kref);
> > extern int copy_ipcs(unsigned long flags, struct task_struct *tsk);
> > extern int unshare_ipcs(unsigned long flags, struct ipc_namespace **ns);
> > -#else

> > -static inline int copy_ipcs(unsigned long flags, struct task_struct *tsk)
> > -{
> > - return 0;
> > -}
> > -#endif
> >
> > static inline struct ipc_namespace *get_ipc_ns(struct ipc_namespace *ns)
> > {
> > -#ifdef CONFIG_IPC_NS
> >   if (ns)
> >   kref_get(&ns->kref);
> > -#endif
> >   return ns;
> > }
> >
> > static inline void put_ipc_ns(struct ipc_namespace *ns)
> > {
> > -#ifdef CONFIG_IPC_NS
> >   kref_put(&ns->kref, free_ipc_ns);
> > -#endif
> > }
> >
> > #endif /* __KERNEL__ */
> > Index: 2.6.20-rc4-mm1/init/Kconfig
> > ===
> > --- 2.6.20-rc4-mm1.orig/init/Kconfig
> > +++ 2.6.20-rc4-mm1/init/Kconfig
> > @@ -138,15 +138,6 @@ config SYSVIPC
> >   section 6.4 of the Linux Programmer's Guide, available from
> >   <http://www.tldp.org/guides.html>.
> >
> > -config IPC_NS
> > - bool "IPC Namespaces"
> > - depends on SYSVIPC
> > - default n
> > - help
> > -   Support ipc namespaces. This allows containers, i.e. virtual
> > -   environments, to use ipc namespaces to provide different ipc
> > -   objects for different servers. If unsure, say N.
> > -
> > config POSIX_MQUEUE
> >   bool "POSIX Message Queues"
> >   depends on NET && EXPERIMENTAL
> > Index: 2.6.20-rc4-mm1/ipc/msg.c
> > ===
> > --- 2.6.20-rc4-mm1.orig/ipc/msg.c
> > +++ 2.6.20-rc4-mm1/ipc/msg.c
> > @@ -87,7 +87,7 @@ static int newque (struct ipc_namespace

> > static int sysvipc_msg_proc_show(struct seq_file *s, void *it);
> > #endif
> >
> > -static void __ipc_init __msg_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> > +static void __msg_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> > {
> >   ns->ids[IPC_MSG_IDS] = ids;
> >   ns->msg_ctlmax = MSGMAX;
> > @@ -96,7 +96,6 @@ static void __ipc_init __msg_init_ns(str
> >   ipc_init_ids(ids, ns->msg_ctlmni);
> > }
> >
> > -#ifdef CONFIG_IPC_NS
> > int msg_init_ns(struct ipc_namespace *ns)
> > {
> >   struct ipc_ids *ids;
> > @@ -128,7 +127,6 @@ void msg_exit_ns(struct ipc_namespace *n
> >   kfree(ns->ids[IPC_MSG_IDS]);
> >   ns->ids[IPC_MSG_IDS] = NULL;
> > }
> > -#endif
> >
> > void __init msg_init(void)
> > {
> > Index: 2.6.20-rc4-mm1/ipc/sem.c
> > ===
> > --- 2.6.20-rc4-mm1.orig/ipc/sem.c
> > +++ 2.6.20-rc4-mm1/ipc/sem.c
> > @@ -122,7 +122,7 @@ static int sysvipc_sem_proc_show(struct
> > #define sc_semopm sem_ctls[2]
> > #define sc_semmni sem_ctls[3]
> >
> > -static void __ipc_init __sem_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> > +static void __sem_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> > {
> >   ns->ids[IPC_SEM_IDS] = ids;
> >   ns->sc_semmsl = SEMMSL;
> > @@ -133,7 +133,6 @@ static void __ipc_init __sem_init_ns(str
> >   ipc_init_ids(ids, ns->sc_semmni);
> > }
> >
> > -#ifdef CONFIG_IPC_NS
> > int sem_init_ns(struct ipc_namespace *ns)
> > {
> >   struct ipc_ids *ids;
> > @@ -165,7 +164,6 @@ void sem_exit_ns(struct ipc_namespace *n
> >   kfree(ns->ids[IPC_SEM_IDS]);
> >   ns->ids[IPC_SEM_IDS] = NULL;

> > }
> > -#endif
> >
> > void __init sem_init (void)
> > {
> > Index: 2.6.20-rc4-mm1/ipc/shm.c
> > ===
> > --- 2.6.20-rc4-mm1.orig/ipc/shm.c
> > +++ 2.6.20-rc4-mm1/ipc/shm.c
> > @@ -67,7 +67,7 @@ static void shm_destroy (struct ipc_name
> > static int sysvipc_shm_proc_show(struct seq_file *s, void *it);
> > #endif
> >
> > -static void __ipc_init __shm_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> > +static void __shm_init_ns(struct ipc_namespace *ns, struct ipc_ids *ids)
> > {
> >   ns->ids[IPC_SHM_IDS] = ids;
> >   ns->shm_ctlmax = SHMMAX;
> > @@ -88,7 +88,6 @@ static void do_shm_rmid(struct ipc_names
> >   shm_destroy(ns, shp);
> > }
> >
> > -#ifdef CONFIG_IPC_NS
> > int shm_init_ns(struct ipc_namespace *ns)
> > {
> >   struct ipc_ids *ids;
> > @@ -120,7 +119,6 @@ void shm_exit_ns(struct ipc_namespace *n
> >   kfree(ns->ids[IPC_SHM_IDS]);
> >   ns->ids[IPC_SHM_IDS] = NULL;
> > }
> > -#endif
> >
> > void __init shm_init (void)
> > {
> > Index: 2.6.20-rc4-mm1/ipc/util.c
> > ===
> > --- 2.6.20-rc4-mm1.orig/ipc/util.c
> > +++ 2.6.20-rc4-mm1/ipc/util.c
> > @@ -51,7 +51,6 @@ struct ipc_namespace init_ipc_ns = {
> >   },
> > };
> >
> > -#ifdef CONFIG_IPC_NS
> > static struct ipc_namespace *clone_ipc_ns(struct ipc_namespace *old_ns)
> > {
> >   int err;
> > @@ -144,7 +143,6 @@ void free_ipc_ns(struct kref *kref)
> >   shm_exit_ns(ns);

```
> >   kfree(ns);
> >  }
> > -#endif
> >
> >  /**
> >   * ipc_init - initialise IPC subsystem
> > @@ -172,7 +170,7 @@ __initcall(ipc_init);
> >   * array itself.
> >   */
> >
> > -void __ipc_init ipc_init_ids(struct ipc_ids* ids, int size)
> > +void ipc_init_ids(struct ipc_ids* ids, int size)
> >  {
> >   int i;
> >
> > Index: 2.6.20-rc4-mm1/ipc/util.h
> > ===================================================================
> > --- 2.6.20-rc4-mm1.orig/ipc/util.h
> > +++ 2.6.20-rc4-mm1/ipc/util.h
> > @@ -41,12 +41,8 @@ struct ipc_ids {
> >  };
> >
> >  struct seq_file;
> > -#ifdef CONFIG_IPC_NS
> > -#define __ipc_init
> > -#else
> > -#define __ipc_init __init
> > -#endif
> > -void __ipc_init ipc_init_ids(struct ipc_ids *ids, int size);
> > +
> > +void ipc_init_ids(struct ipc_ids *ids, int size);
> >  #ifdef CONFIG_PROC_FS
> >  void __init ipc_init_proc_interface(const char *path, const char *header,
> >    int ids, int (*show)(struct seq_file *, void *));
> > Index: 2.6.20-rc4-mm1/kernel/fork.c
> > ===================================================================
> > --- 2.6.20-rc4-mm1.orig/kernel/fork.c
> > +++ 2.6.20-rc4-mm1/kernel/fork.c
> > @@ -1595,16 +1595,6 @@ static int unshare_semundo(unsigned long
> >   return 0;
> >  }
> >
> > -#ifndef CONFIG_IPC_NS
> > -static inline int unshare_ipcs(unsigned long flags, struct ipc_namespace **ns)
> > -{
> > - if (flags & CLONE_NEWIPC)
> > -  return -EINVAL;
> > -
```

> > - return 0;
> > -}
> > -#endif
> > -
> > /*
> >   * unshare allows a process to 'unshare' part of the process
> >   * context which was originally shared using clone. copy_*
> > _____
> > 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

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