
Subject: [RFC][PATCH v2 6/7] Recomputing msgmni on ipc namespace creation/removal

Posted by [Nadia Derby](#) on Thu, 31 Jan 2008 13:40:24 GMT

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

[PATCH 06/07]

This patch introduces a notification mechanism that aims at recomputing msgmni each time an ipc namespace is created or removed.

The ipc namespace notifier chain already defined for memory hotplug management is used for that purpose too.

Each time a new ipc namespace is allocated or an existing ipc namespace is removed, the ipcns notifier chain is notified. The callback routine for each registered ipc namespace is then activated in order to recompute msgmni for that namespace.

Signed-off-by: Nadia Derby <Nadia.Derbey@bull.net>

include/linux/ipc.h | 25 ++-----
ipc/Makefile | 3 +-
ipc/ipcns_notifier.c | 2 ++
ipc/util.c | 12 ++++++++
4 files changed, 17 insertions(+), 25 deletions(-)

Index: linux-2.6.24/include/linux/ipc.h

```
=====
--- linux-2.6.24.orig/include/linux/ipc.h 2008-01-31 10:49:07.000000000 +0100
+++ linux-2.6.24/include/linux/ipc.h 2008-01-31 11:46:32.000000000 +0100
@@ -81,9 +81,7 @@ struct ipc_kludge {

#include <linux/kref.h>
#include <linux/spinlock.h>
-#ifdef CONFIG_MEMORY_HOTPLUG
#include <linux/notifier.h>
-#endif /* CONFIG_MEMORY_HOTPLUG */

#define IPCMNI 32768 /* <= MAX_INT limit for ipc arrays (including sysctl changes) */

@@ -92,6 +90,8 @@ struct ipc_kludge {
 * ipc namespace events
 */
#define IPCNS_MEMCHANGED 0x00000001 /* Notify lowmem size changed */
+#define IPCNS_CREATED 0x00000002 /* Notify new ipc namespace created */
+#define IPCNS_REMOVED 0x00000003 /* Notify ipc namespace removed */
```

```

#define IPCNS_CALLBACK_PRI 0

@@ -130,9 +130,7 @@ struct ipc_namespace {
    int shm_ctlmni;
    int shm_tot;

-#ifdef CONFIG_MEMORY_HOTPLUG
    struct notifier_block ipcns_nb;
-#endif
};

extern struct ipc_namespace init_ipc_ns;
@@ -143,29 +141,10 @@ extern atomic_t nr_ipc_ns;
extern void free_ipc_ns(struct kref *kref);
extern struct ipc_namespace *copy_ipcs(unsigned long flags,
    struct ipc_namespace *ns);
-#ifdef CONFIG_MEMORY_HOTPLUG
-
extern int register_ipcns_notifier(struct ipc_namespace *);
extern int unregister_ipcns_notifier(struct ipc_namespace *);
extern int ipcns_notify(unsigned long);

-#else /* CONFIG_MEMORY_HOTPLUG */
-
-static inline int register_ipcns_notifier(struct ipc_namespace *ipcns)
-{
- return 0;
-}
-static inline int unregister_ipcns_notifier(struct ipc_namespace *ipcns)
-{
- return 0;
-}
-static inline int ipcns_notify(unsigned long ev)
-{
- return 0;
-}
-
-#endif /* CONFIG_MEMORY_HOTPLUG */
-
#else
#define INIT_IPC_NS(ns)
static inline struct ipc_namespace *copy_ipcs(unsigned long flags,
Index: linux-2.6.24/ipc/ipcns_notifier.c
=====
--- linux-2.6.24.orig/ipc/ipcns_notifier.c 2008-01-31 10:58:11.000000000 +0100
+++ linux-2.6.24/ipc/ipcns_notifier.c 2008-01-31 11:48:32.000000000 +0100
@@ -27,6 +27,8 @@ static int ipcns_callback(struct notifie

```

```

switch (action) {
case IPCNS_MEMCHANGED: /* amount of lowmem has changed */
+ case IPCNS_CREATED:
+ case IPCNS_REMOVED:
/*
 * It's time to recompute msgmni
 */

```

Index: linux-2.6.24/ipc/Makefile

```

=====
--- linux-2.6.24.orig/ipc/Makefile 2008-01-31 10:59:39.000000000 +0100
+++ linux-2.6.24/ipc/Makefile 2008-01-31 11:49:13.000000000 +0100
@@ -3,8 +3,7 @@
#

```

```

obj-$(CONFIG_SYSVIPC_COMPAT) += compat.o
-obj_mem-$(CONFIG_MEMORY_HOTPLUG) += ipcns_notifier.o
-obj-$(CONFIG_SYSVIPC) += util.o msgutil.o msg.o sem.o shm.o $(obj_mem-y)
+obj-$(CONFIG_SYSVIPC) += util.o msgutil.o msg.o sem.o shm.o ipcns_notifier.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)
Index: linux-2.6.24/ipc/util.c

```

```

=====
--- linux-2.6.24.orig/ipc/util.c 2008-01-31 11:41:01.000000000 +0100
+++ linux-2.6.24/ipc/util.c 2008-01-31 12:00:46.000000000 +0100
@@ -117,6 +117,12 @@ static struct ipc_namespace *clone_ipc_n
if (err)
goto err_shm;

```

```

+ /*
+ * msgmni has already been computed for the new ipc ns.
+ * Thus, do the ipcns creation notification before registering that
+ * new ipcns in the chain.
+ */
+ ipcns_notify(IPCNS_CREATED);
+ register_ipcns_notifier(ns);

kref_init(&ns->kref);
@@ -168,6 +174,12 @@ void free_ipc_ns(struct kref *kref)
shm_exit_ns(ns);
kfree(ns);
atomic_dec(&nr_ipc_ns);
+
+ /*
+ * Do the ipcns removal notification after decrementing nr_ipc_ns in
+ * order to have a correct value when recomputing msgmni.
+ */

```

```
+ ipcns_notify(IPCNS_REMOVED);  
}
```

```
/**
```

```
--
```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
