
Subject: [RFC][PATCH 3/4] IPC: use the target ID specified in procs
Posted by [Nadia Derby](#) on Fri, 28 Mar 2008 09:53:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

[PATCH 03/04]

This patch makes use of the target id specified by a previous write into /proc/self/next_id as the id to use to allocate the next IPC object.

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

include/linux/sysids.h | 7 +++++++
ipc/util.c | 39 +++++++++++++++++++++++++++++++++++++-----
kernel/nextid.c | 2 +-
3 files changed, 39 insertions(+), 9 deletions(-)

Index: linux-2.6.25-rc3-mm1/include/linux/sysids.h

```
=====
--- linux-2.6.25-rc3-mm1.orig/include/linux/sysids.h 2008-03-28 08:19:49.000000000 +0100
+++ linux-2.6.25-rc3-mm1/include/linux/sysids.h 2008-03-28 08:21:02.000000000 +0100
@@ -37,7 +37,14 @@ struct sys_id {
     long *blocks[0];
 };

+#define next_ipcid(tsk) ((tsk)->next_id \
+ ? ((tsk)->next_id->nids \
+ ? ID_AT((tsk)->next_id, 0) \
+ : -1) \
+ : -1)
+
extern ssize_t get_nextid(struct task_struct *, char *);
extern int set_nextid(struct task_struct *, char *);
+extern void id_blocks_free(struct sys_id *);
```

```
#endif /* _LINUX_SYSIDS_H */
```

Index: linux-2.6.25-rc3-mm1/kernel/nextid.c

```
=====
--- linux-2.6.25-rc3-mm1.orig/kernel/nextid.c 2008-03-28 08:20:52.000000000 +0100
+++ linux-2.6.25-rc3-mm1/kernel/nextid.c 2008-03-28 08:21:02.000000000 +0100
@@ -49,7 +49,7 @@ out_undo_partial_alloc:
     return NULL;
 }

-static void id_blocks_free(struct sys_id *ids)
+void id_blocks_free(struct sys_id *ids)
{
    if (ids == NULL)
```

return;

Index: linux-2.6.25-rc3-mm1/ipc/util.c

=====

--- linux-2.6.25-rc3-mm1.orig/ipc/util.c 2008-03-28 08:19:49.000000000 +0100

+++ linux-2.6.25-rc3-mm1/ipc/util.c 2008-03-28 08:21:02.000000000 +0100

@@ -260,6 +260,7 @@ int ipc_get_maxid(struct ipc_ids *ids)

int ipc_addid(struct ipc_ids* ids, struct kern_ipc_perm* new, int size)

{
int id, err;
+ int next_id;

if (size > IPCMNI)

size = IPCMNI;

@@ -267,20 +268,42 @@ int ipc_addid(struct ipc_ids* ids, struc

if (ids->in_use >= size)

return -ENOSPC;

- err = idr_get_new(&ids->ipcs_idr, new, &id);

- if (err)

- return err;

+ next_id = next_ipcid(current);

+ if (next_id >= 0) {

+ /* There is a target id specified, try to use it */

+ int new_lid = next_id % SEQ_MULTIPLIER;

+

+ if (next_id !=

+ (new_lid + (next_id / SEQ_MULTIPLIER) * SEQ_MULTIPLIER))

+ return -EINVAL;

+

+ err = idr_get_new_above(&ids->ipcs_idr, new, new_lid, &id);

+ if (err)

+ return err;

+ if (id != new_lid) {

+ idr_remove(&ids->ipcs_idr, id);

+ return -EBUSY;

+ }

+

+ new->id = next_id;

+ new->seq = next_id / SEQ_MULTIPLIER;

+ id_blocks_free(current->next_id);

+ } else {

+ err = idr_get_new(&ids->ipcs_idr, new, &id);

+ if (err)

+ return err;

+

+ new->seq = ids->seq++;

+ if (ids->seq > ids->seq_max)

+ ids->seq = 0;

```
+ new->id = ipc_buildid(id, new->seq);
+ }

ids->in_use++;

new->cuid = new->uid = current->euid;
new->gid = new->cgid = current->egid;

- new->seq = ids->seq++;
- if(ids->seq > ids->seq_max)
-   ids->seq = 0;
-
- new->id = ipc_buildid(id, new->seq);
  spin_lock_init(&new->lock);
  new->deleted = 0;
  rcu_read_lock();

--
```

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