
Subject: [patch -mm 15/17] pid namespace: add unshare
Posted by [Cedric Le Goater](#) on Tue, 05 Dec 2006 10:28:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Cedric Le Goater <clg@fr.ibm.com>

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

```
include/linux/pid_namespace.h | 2 +
kernel/nsproxy.c              | 25 ++++++
kernel/pid.c                   | 44 ++++++
3 files changed, 68 insertions(+), 3 deletions(-)
```

Index: 2.6.19-rc6-mm2/include/linux/pid_namespace.h

=====

```
--- 2.6.19-rc6-mm2.orig/include/linux/pid_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/pid_namespace.h
@@ -29,6 +29,8 @@ static inline void get_pid_ns(struct pid
    kref_get(&ns->kref);
}
```

```
+extern int unshare_pid_ns(unsigned long unshare_ns_flags,
+ struct pid_namespace **new_pid);
extern int copy_pid_ns(int flags, struct task_struct *tsk);
extern void free_pid_ns(struct kref *kref);
```

Index: 2.6.19-rc6-mm2/kernel/pid.c

=====

```
--- 2.6.19-rc6-mm2.orig/kernel/pid.c
+++ 2.6.19-rc6-mm2/kernel/pid.c
@@ -361,6 +361,50 @@ struct pid *find_ge_pid(int nr)
}
EXPORT_SYMBOL_GPL(find_get_pid);
```

```
+static struct pid_namespace *clone_pid_ns(struct pid_namespace *old_ns)
+{
+ struct pid_namespace *ns;
+ int i;
+
+ ns = kmalloc(sizeof(struct pid_namespace), GFP_KERNEL);
+ if (!ns)
+ return ns;
+
+ kref_init(&ns->kref);
+
+ atomic_set(&ns->pidmap[0].nr_free, BITS_PER_PAGE - 1);
+ ns->pidmap[0].page = kzalloc(PAGE_SIZE, GFP_KERNEL);
+ if (!ns->pidmap[0].page) {
```

```

+ kfree(ns);
+ return NULL;
+ }
+
+ set_bit(0, ns->pidmap[0].page);
+
+ for (i = 1; i < PIDMAP_ENTRIES; i++) {
+ atomic_set(&ns->pidmap[i].nr_free, BITS_PER_PAGE);
+ ns->pidmap[i].page = NULL;
+ }
+ ns->last_pid = 0;
+ ns->child_reaper = current;
+ return ns;
+}
+
+int unshare_pid_ns(unsigned long unshare_ns_flags,
+ struct pid_namespace **new_pid)
+{
+ if (unshare_ns_flags & NS_PID) {
+ if (!capable(CAP_SYS_ADMIN))
+ return -EPERM;
+
+ *new_pid = clone_pid_ns(current->nsproxy->pid_ns);
+ if (!*new_pid)
+ return -ENOMEM;
+ }
+
+ return 0;
+}
+
+int copy_pid_ns(int flags, struct task_struct *tsk)
+{
+ struct pid_namespace *old_ns = tsk->nsproxy->pid_ns;

```

Index: 2.6.19-rc6-mm2/kernel/nsproxy.c

=====

--- 2.6.19-rc6-mm2.orig/kernel/nsproxy.c

+++ 2.6.19-rc6-mm2/kernel/nsproxy.c

@@ -324,6 +324,11 @@ static int switch_ns(int id, unsigned lo

```

    put_ipc_ns(new_ns->ipc_ns);
    new_ns->ipc_ns = get_ipc_ns(ns->ipc_ns);
}

```

```

+ if (flags & NS_PID) {
+ get_pid_ns(ns->pid_ns);
+ put_pid_ns(new_ns->pid_ns);
+ new_ns->pid_ns = ns->pid_ns;
+ }

```

```

    out_ns:
    put_nsproxy(ns);

```

```

}
@@ -440,6 +445,7 @@ asmlinkage long sys_unshare_ns(unsigned
    struct mnt_namespace *mnt, *new_mnt = NULL;
    struct uts_namespace *uts, *new_uts = NULL;
    struct ipc_namespace *ipc, *new_ipc = NULL;
+ struct pid_namespace *pid, *new_pid = NULL;
    unsigned long unshare_flags = 0;

    /* Return -EINVAL for all unsupported flags */
@@ -467,16 +473,19 @@ asmlinkage long sys_unshare_ns(unsigned
    if ((err = unshare_ipcs(unshare_flags, &new_ipc)))
        goto bad_unshare_ns_cleanup_uts;

- if (new_mnt || new_uts || new_ipc) {
+ if ((err = unshare_pid_ns(unshare_ns_flags, &new_pid)))
+ goto bad_unshare_ns_cleanup_ipc;
+
+ if (new_mnt || new_uts || new_ipc || new_pid) {
    old_nsproxy = current->nsproxy;
    new_nsproxy = dup_namespaces(old_nsproxy);
    if (!new_nsproxy) {
        err = -ENOMEM;
- goto bad_unshare_ns_cleanup_ipc;
+ goto bad_unshare_ns_cleanup_pid;
    }
}

- if (new_fs || new_mnt || new_uts || new_ipc) {
+ if (new_fs || new_mnt || new_uts || new_ipc || new_pid) {

    task_lock(current);

@@ -509,12 +518,22 @@ asmlinkage long sys_unshare_ns(unsigned
    new_ipc = ipc;
}

+ if (new_pid) {
+ pid = current->nsproxy->pid_ns;
+ current->nsproxy->pid_ns = new_pid;
+ new_pid = pid;
+ }
+
    task_unlock(current);
}

if (new_nsproxy)
    put_nsproxy(new_nsproxy);

```

```
+bad_unshare_ns_cleanup_pid:  
+ if (new_pid)  
+  put_pid_ns(new_pid);  
+  
bad_unshare_ns_cleanup_ipc:  
  if (new_ipc)  
    put_ipc_ns(new_ipc);
```

--

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