
Subject: [PATCH 3/4] Virtualization/containers: UID hash
Posted by [Kirill Korotaev](#) on Mon, 06 Feb 2006 22:15:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch virtualizes UID hash, so that processes in container can use it's own UID set.
Can be done as an option if some virtualization solutions do not require it.

Signed-Off-By: Kirill Korotaev <dev@openvz.org>

Kirill

```
--- ./include/linux/container.h.uids 2006-02-06 23:46:40.000000000 +0300
+++ ./include/linux/container.h 2006-02-07 00:05:33.000000000 +0300
@@ -6,11 +6,14 @@
#include <asm/atomic.h>

struct task_struct;
+struct list_head;

struct container {
    u32 id;
    struct task_struct *init_task;
    atomic_t refcnt;
+
+ struct list_head *c_uid_hash;
};

extern struct container init_container;
--- ./kernel/user.c.uids 2006-02-06 22:15:06.000000000 +0300
+++ ./kernel/user.c 2006-02-06 23:58:06.000000000 +0300
@@ -14,6 +14,7 @@
#include <linux/bitops.h>
#include <linux/key.h>
#include <linux/interrupt.h>
+#include <linux/container.h>

/*
 * UID task count cache, to get fast user lookup in "alloc_uid"
@@ -24,7 +25,12 @@
#define UIDHASH_SZ (1 << UIDHASH_BITS)
#define UIDHASH_MASK (UIDHASH_SZ - 1)
#define __uidhashfn(uid) (((uid >> UIDHASH_BITS) + uid) & UIDHASH_MASK)
+
+ #ifdef CONFIG_CONTAINER
+ #define uidhashentry(uid) (econtainer()->c_uid_hash + __uidhashfn((uid)))
+ #else
+ #define uidhashentry(uid) (uidhash_table + __uidhashfn((uid)))
```

```
+#endif
```

```
static kmem_cache_t *uid_cache;
static struct list_head uidhash_table[UIDHASH_SZ];
@@ -200,6 +206,9 @@ static int __init uid_cache_init(void)

/* Insert the root user immediately (init already runs as root) */
spin_lock_irq(&uidhash_lock);
#ifdef CONFIG_CONTAINER
+ init_container.c_uid_hash = uidhash_table;
#endif
uid_hash_insert(&root_user, uidhashentry(0));
spin_unlock_irq(&uidhash_lock);
```
