
Subject: [RFC][PATCH 2/5] Virtualization/containers: UIDs

Posted by [dev](#) on Fri, 03 Feb 2006 17:01:30 GMT

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

The simplest virtualization of UID hashes, just to demonstrate our approach. Each container has it's own set of uids and should simply allocate hash/initialize it on startup.

Kirill

```
--- ./include/linux/vps_info.h.vps_uid_hash 2006-02-03 16:49:26.000000000 +0300
```

```
+++ ./include/linux/vps_info.h 2006-02-03 16:49:51.000000000 +0300
```

```
@@ -5,11 +5,14 @@
```

```
#include <asm/atomic.h>
```

```
struct task_struct;
```

```
+struct list_head;
```

```
struct vps_info {
```

```
    u32 id;
```

```
    struct task_struct *init_task;
```

```
    atomic_t refcnt;
```

```
+
```

```
+ struct list_head *vps_uid_hash;
```

```
};
```

```
extern struct vps_info host_vps_info;
```

```
--- ./kernel/user.c.vps_uid_hash 2006-02-03 16:49:08.000000000 +0300
```

```
+++ ./kernel/user.c 2006-02-03 16:49:51.000000000 +0300
```

```
@@ -14,6 +14,7 @@
```

```
#include <linux/bitops.h>
```

```
#include <linux/key.h>
```

```
#include <linux/interrupt.h>
```

```
+#include <linux/vps_info.h>
```

```
/*
```

```
 * UID task count cache, to get fast user lookup in "alloc_uid"
```

```
@@ -24,7 +25,8 @@
```

```
#define UIDHASH_SZ (1 << UIDHASH_BITS)
```

```
#define UIDHASH_MASK (UIDHASH_SZ - 1)
```

```
#define __uidhashfn(uid) (((uid >> UIDHASH_BITS) + uid) & UIDHASH_MASK)
```

```
-#define uidhashentry(uid) (uidhash_table + __uidhashfn((uid)))
```

```
+#define uidhashentry(uid) (current_vps()->vps_uid_hash + \
```

```
+    __uidhashfn((uid)))
```

```
static kmem_cache_t *uid_cachep;
```

```
static struct list_head uidhash_table[UIDHASH_SZ];
```

```
@@ -200,6 +202,7 @@
```

```
/* Insert the root user immediately (init already runs as root) */
spin_lock_irq(&uidhash_lock);
+ host_vps_info.vps_uid_hash = uidhash_table;
  uid_hash_insert(&root_user, uidhashentry(0));
  spin_unlock_irq(&uidhash_lock);
```
