
Subject: [PATCH 4/5] Move the user namespace under the option

Posted by [Pavel Emelianov](#) on Mon, 01 Oct 2007 15:40:13 GMT

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

We currently have a CONFIG_USER_NS option. Just rename it into CONFIG_NAMESPACES_EXPERIMENTAL and move the init_user_ns into user.c file to make the kernel compile and work without the namespaces support.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

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

```
diff --git a/include/linux/user_namespace.h b/include/linux/user_namespace.h
index b5f41d4..dda160c 100644
```

```
--- a/include/linux/user_namespace.h
+++ b/include/linux/user_namespace.h
@@ -17,7 +17,7 @@ struct user_namespace {
```

```
extern struct user_namespace init_user_ns;
```

```
+#ifndef CONFIG_USER_NS
+#ifdef CONFIG_NAMESPACES_EXPERIMENTAL
```

```
static inline struct user_namespace *get_user_ns(struct user_namespace *ns)
{
```

```
diff --git a/init/Kconfig b/init/Kconfig
index 6bb603a..67b1e41 100644
```

```
--- a/init/Kconfig
+++ b/init/Kconfig
@@ -206,15 +206,6 @@ config TASK_IO_ACCOUNTING
```

Say N if unsure.

```
-config USER_NS
- bool "User Namespaces (EXPERIMENTAL)"
- default n
- depends on EXPERIMENTAL
- help
- Support user namespaces. This allows containers, i.e.
- vservers, to use user namespaces to provide different
- user info for different servers. If unsure, say N.
-
```

```
config AUDIT
bool "Auditing support"
depends on NET
```

```
diff --git a/kernel/Makefile b/kernel/Makefile
```

index 76f782f..5817bfe 100644

--- a/kernel/Makefile

+++ b/kernel/Makefile

@@ -4,7 +4,7 @@

```
obj-y = sched.o fork.o exec_domain.o panic.o printk.o profile.o \  
  exit.o itimer.o time.o softirq.o resource.o \  
- sysctl.o capability.o ptrace.o timer.o user.o user_namespace.o \  
+ sysctl.o capability.o ptrace.o timer.o user.o \  
  signal.o sys.o kmod.o workqueue.o pid.o \  
  rcupdate.o extable.o params.o posix-timers.o \  
  kthread.o wait.o kfifo.o sys_ni.o posix-cpu-timers.o mutex.o \  
@@ -50,6 +49,7 @@ obj-$(CONFIG_AUDITSYSCALL) += auditsc.o  
obj-$(CONFIG_KPROBES) += kprobes.o  
obj-$(CONFIG_KGDB) += kgdb.o  
obj-$(CONFIG_NAMESPACES) += utsname.o  
+obj-$(CONFIG_NAMESPACES_EXPERIMENTAL) += user_namespace.o  
obj-$(CONFIG_SYSFS) += ksysfs.o  
obj-$(CONFIG_DETECT_SOFTLOCKUP) += softlockup.o  
obj-$(CONFIG_GENERIC_HARDIRQS) += irq/  
diff --git a/kernel/user.c b/kernel/user.c
```

index 80fbe56..2f62cf4 100644

--- a/kernel/user.c

+++ b/kernel/user.c

@@ -17,6 +17,15 @@

#include <linux/module.h>

#include <linux/user_namespace.h>

```
+struct user_namespace init_user_ns = {  
+ .kref = {  
+ .refcount = ATOMIC_INIT(2),  
+ },  
+ .root_user = &root_user,  
+};  
+  
+EXPORT_SYMBOL_GPL(init_user_ns);  
+  
/*  
 * UID task count cache, to get fast user lookup in "alloc_uid"  
 * when changing user ID's (ie setuid() and friends).  
@@ -242,6 +251,7 @@ void switch_uid(struct user_struct *new_  
  suid_keys(current);  
}
```

```
+#ifdef CONFIG_NAMESPACES_EXPERIMENTAL
```

```
void release_uids(struct user_namespace *ns)
```

```
{  
  int i;
```

```

@@ -266,6 +276,7 @@ void release_uids(struct user_namespace

    free_uid(ns->root_user);
}
+#endif

static int __init uid_cache_init(void)
{
diff --git a/kernel/user_namespace.c b/kernel/user_namespace.c
index 7af90fc..4c90062 100644
--- a/kernel/user_namespace.c
+++ b/kernel/user_namespace.c
@@ -10,17 +10,6 @@
#include <linux/nsproxy.h>
#include <linux/user_namespace.h>

-struct user_namespace init_user_ns = {
- .kref = {
- .refcount = ATOMIC_INIT(2),
- },
- .root_user = &root_user,
-};
-
-EXPORT_SYMBOL_GPL(init_user_ns);
-
-#ifdef CONFIG_USER_NS
-
/*
 * Clone a new ns copying an original user ns, setting refcount to 1
 * @old_ns: namespace to clone
@@ -84,5 +73,3 @@ void free_user_ns(struct kref *kref)
    release_uids(ns);
    kfree(ns);
}
-
-#endif /* CONFIG_USER_NS */

```
