
Subject: [PATCH] kthread: Spontaneous exit support
Posted by [ebiederm](#) on Mon, 23 Apr 2007 03:12:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch implements the kthread helper functions kthread_start and kthread_end which make it simple to support a kernel thread that may decided to exit on it's own before we request it to. It is still assumed that eventually we will get around to requesting that the kernel thread stop.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
include/linux/kthread.h | 23 ++++++
kernel/kthread.c        | 18 ++++++
2 files changed, 41 insertions(+), 0 deletions(-)
```

```
diff --git a/include/linux/kthread.h b/include/linux/kthread.h
```

```
index a8ea31d..4f1eff1 100644
```

```
--- a/include/linux/kthread.h
```

```
+++ b/include/linux/kthread.h
```

```
@@ -28,6 +28,29 @@ struct task_struct *kthread_create(int (*threadfn)(void *data),
```

```
void kthread_bind(struct task_struct *k, unsigned int cpu);
```

```
int kthread_stop(struct task_struct *k);
```

```
+/**
```

```
+ * kthread_start - create and wake a thread.
```

```
+ * @threadfn: the function to run until kthread_should_stop().
```

```
+ * @data: data ptr for @threadfn.
```

```
+ * @namefmt: printf-style name for the thread.
```

```
+ *
```

```
+ * Description: Convenient wrapper for kthread_create() followed by
```

```
+ * get_task_struct() and wake_up_process. kthread_start should be paired
```

```
+ * with kthread_end() so we don't leak task structs.
```

```
+ *
```

```
+ * Returns the kthread or ERR_PTR(-ENOMEM).
```

```
+ */
```

```
+#define kthread_start(threadfn, data, namefmt, ...) \
```

```
+({ \
```

```
+ struct task_struct *__k \
```

```
+ = kthread_create(threadfn, data, namefmt, ## __VA_ARGS__); \
```

```
+ if (!IS_ERR(__k)) { \
```

```
+ get_task_struct(__k); \
```

```
+ wake_up_process(__k); \
```

```
+ } \
```

```
+ __k; \
```

```
+})
```

```
+int kthread_end(struct task_struct *k);
```

```

static inline int __kthread_should_stop(struct task_struct *tsk)
{
diff --git a/kernel/kthread.c b/kernel/kthread.c
index 9b3c19f..d6d63c6 100644
--- a/kernel/kthread.c
+++ b/kernel/kthread.c
@@ -179,6 +179,24 @@ int kthread_stop(struct task_struct *tsk)
}
EXPORT_SYMBOL(kthread_stop);

+/**
+ * kthread_end - signal a kthread and wait for it to exit.
+ * @task: The kthread to end.
+ *
+ * Description: Convenient wrapper for kthread_stop() followed by
+ * put_task_struct(). Returns the kthread exit code.
+ *
+ * kthread_start()/kthread_end() can handle kthread that spontaneously exit
+ * before the kthread is requested to terminate.
+ */
+int kthread_end(struct task_struct *task)
+{
+ int ret;
+ ret = kthread_stop(task);
+ put_task_struct(task);
+ return ret;
+}
+EXPORT_SYMBOL(kthread_end);

static __init void kthreadd_setup(void)
{
--
1.5.0.g53756

```

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