

# [PATCH 1/2] uml: fix not\_dead\_yet when directory is in bad state

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-04/msg06732.html>

---

- *From:* Paolo 'Blaisorblade' Giarrusso <[blaisorblade@xxxxxxxx](mailto:blaisorblade@xxxxxxxx)>
  - *Date:* Sun, 30 Apr 2006 17:36:35 +0200
- 

From: Paolo 'Blaisorblade' Giarrusso <[blaisorblade@xxxxxxxx](mailto:blaisorblade@xxxxxxxx)>

The bug occurred to me when a UML left an empty `~/uml/Sarge-norm` folder – when trying to reuse `not_dead_yet()` failed one of its check. The comment says that's ok and means that we can take the directory, but while normally `not_dead_yet()` removes it and returns 0 (i.e. go on, use this), on failure it returns 0 but forgets to remove it.

The fix is to remove it anytime we're going to return 0.

But since "not\_dead\_yet" didn't make the interface so clear, causing this bug, and I couldn't find a convenient name for the mix of things it did, I split it into two parts:

`is_umdir_used()` – returns a boolean, contains all checks of `not_dead_yet()`  
`umdir_take_if_dead` – tries to remove the dir unless it's used – returns whether it removed it, that is we now own it.

With this changes the control flow is IMHO a bit clearer and needs less comment for control flow.

Signed-off-by: Paolo 'Blaisorblade' Giarrusso <[blaisorblade@xxxxxxxx](mailto:blaisorblade@xxxxxxxx)>

---

```
arch/um/os-Linux/umid.c | 48 ++++++-----  
1 files changed, 29 insertions(+), 19 deletions(-)
```

```
diff --git a/arch/um/os-Linux/umid.c b/arch/um/os-Linux/umid.c  
index 34bfc1b..7c53be1 100644  
--- a/arch/um/os-Linux/umid.c  
+++ b/arch/um/os-Linux/umid.c  
@@ -103,9 +103,10 @@ static int actually_do_remove(char *dir)  
* something other than UML sticking stuff in the directory  
* this boot racing with a shutdown of the other UML  
* In any of these cases, the directory isn't useful for anything else.  
+ *  
+ * Boolean return: 1 if in use, 0 otherwise.  
*/  
-
```

[PATCH 1/2] uml: fix not\_dead\_yet when directory is in bad state

```
-static int not_dead_yet(char *dir)
+static inline int is_umdir_used(char *dir)
{
char file[strlen(uml_dir) + UMID_LEN + sizeof("/pid\0")];
char pid[sizeof("nnnnn\0")], *end;
@@ -113,7 +114,7 @@ static int not_dead_yet(char *dir)

n = snprintf(file, sizeof(file), "%s/pid", dir);
if(n >= sizeof(file)){
- printk("not_dead_yet - pid filename too long\n");
+ printk("is_umdir_used - pid filename too long\n");
err = -E2BIG;
goto out;
}
@@ -123,7 +124,7 @@ static int not_dead_yet(char *dir)
if(fd < 0) {
fd = -errno;
if(fd != -ENOENT){
- printk("not_dead_yet : couldn't open pid file '%s', "
+ printk("is_umdir_used : couldn't open pid file '%s', "
"err = %d\n", file, -fd);
}
goto out;
@@ -132,18 +133,18 @@ static int not_dead_yet(char *dir)
err = 0;
n = read(fd, pid, sizeof(pid));
if(n < 0){
- printk("not_dead_yet : couldn't read pid file '%s', "
+ printk("is_umdir_used : couldn't read pid file '%s', "
"err = %d\n", file, errno);
goto out_close;
} else if(n == 0){
- printk("not_dead_yet : couldn't read pid file '%s', "
+ printk("is_umdir_used : couldn't read pid file '%s', "
"0-byte read\n", file);
goto out_close;
}

p = strtoul(pid, &end, 0);
if(end == pid){
- printk("not_dead_yet : couldn't parse pid file '%s', "
+ printk("is_umdir_used : couldn't parse pid file '%s', "
"errno = %d\n", file, errno);
goto out_close;
}
@@ -153,19 +154,32 @@ static int not_dead_yet(char *dir)
return 1;
}

- err = actually_do_remove(dir);
- if(err)
```

[PATCH 1/2] uml: fix not\_dead\_yet when directory is in bad state

```
- printk("not_dead_yet - actually_do_remove failed with "
- "err = %d\n", err);
-
- return err;
-
out_close:
close(fd);
out:
return 0;
}

+/*
+ * Try to remove the directory @dir unless it's in use.
+ * Precondition: @dir exists.
+ * Returns 0 for success, < 0 for failure in removal or if the directory is in
+ * use.
+ */
+static int umdir_take_if_dead(char *dir)
+{
+ int ret;
+ if (is_umdir_used(dir))
+ return -EEXIST;
+
+ ret = actually_do_remove(dir);
+ if (ret) {
+ printk("is_umdir_used - actually_do_remove failed with "
+ "err = %d\n", ret);
+ }
+ return ret;
+}
+
static void __init create_pid_file(void)
{
char file[strlen(uml_dir) + UMID_LEN + sizeof("/pid\0")];
@@ -244,11 +258,7 @@ int __init make_umid(void)
if(err != -EEXIST)
goto err;

- /* 1 -> this umid is already in use
- * < 0 -> we couldn't remove the umid directory
- * In either case, we can't use this umid, so return -EEXIST.
- */
- if(not_dead_yet(tmp) != 0)
+ if(umdir_take_if_dead(tmp) < 0)
goto err;

err = mkdir(tmp, 0777);
-
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>

[PATCH 1/2] uml: fix not\_dead\_yet when directory is in bad state

Please read the FAQ at <http://www.tux.org/lkml/>