
Subject: Re: [PATCH] ia64 sn xpc: Convert to use kthread API.

Posted by [Dean Nelson](#) on Thu, 26 Apr 2007 20:00:47 GMT

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

On Thu, Apr 19, 2007 at 01:58:44AM -0600, Eric W. Biederman wrote:

> From: Eric W. Biederman <ebiederm@xmission.com>

>

> This patch starts the xpc kernel threads using kthread_run

> not a combination of kernel_thread and daemonize. Resulting

> in slightly simpler and more maintainable code.

>

> Cc: Jes Sorensen <jes@sgi.com>

> Cc: Tony Luck <tony.luck@intel.com>

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

> ---

> arch/ia64/sn/kernel/xpc_main.c | 31 ++++++++-----

> 1 files changed, 13 insertions(+), 18 deletions(-)

Acked-by: Dean Nelson <dcn@sgi.com>

Andrew, I've tested Eric's patch in conjunction with a fix from Christoph Lameter, which you've already got (it cleaned up a couple of compiler errors), and the following patch that I'd like added (it cleans up a couple of compiler warning errors and makes a few cosmetic changes).

Thanks,
Dean

Signed-off-by: Dean Nelson <dcn@sgi.com>

Index: mm-tree/arch/ia64/sn/kernel/xpc_main.c

```
=====
--- mm-tree.orig/arch/ia64/sn/kernel/xpc_main.c 2007-04-25 14:04:51.701213426 -0500
+++ mm-tree/arch/ia64/sn/kernel/xpc_main.c 2007-04-26 06:29:02.447330438 -0500
@@ -568,7 +568,6 @@ @@
```

```
task = kthread_run(xpc_activating, (void *) ((u64) partid),
    "xpc%02d", partid);
```

```
-
```

```
if (unlikely(IS_ERR(task))) {
    spin_lock_irqsave(&part->act_lock, irq_flags);
    part->act_state = XPC_P_INACTIVE;
```

```
@@ -808,7 +807,6 @@ @@
```

```
int ignore_disconnecting)
```

```
{
```

```
    unsigned long irq_flags;
```

```
- pid_t pid;
```

```

u64 args = XPC_PACK_ARGS(ch->partid, ch->number);
struct xpc_partition *part = &xpc_partitions[ch->partid];
struct task_struct *task;
@@ -840,7 +838,7 @@
(void) xpc_part_ref(part);
xpc_msgqueue_ref(ch);

- task = kthread_run(xpc_daemonize_kthread, args,
+ task = kthread_run(xpc_daemonize_kthread, (void *) args,
    "xpc%02dc%d", ch->partid, ch->number);
if (IS_ERR(task)) {
    /* the fork failed */
@@ -1381,7 +1379,8 @@
    * activate based on info provided by SAL. This new thread is short
    * lived and will exit once discovery is complete.
    */
- task = kthread_run(xpc_initiate_discovery, NULL, XPC_DISCOVERY_THREAD_NAME);
+ task = kthread_run(xpc_initiate_discovery, NULL,
+    XPC_DISCOVERY_THREAD_NAME);
if (IS_ERR(task)) {
    dev_err(xpc_part, "failed while forking discovery thread\n");

```

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
