Subject: Re: [PATCH] ibmphp: Convert to use the kthreads API
Posted by Christoph Hellwig on Sun, 22 Apr 2007 12:09:50 GMT
View Forum Message <> Reply to Message

On Thu, Apr 19, 2007 at 12:55:30AM -0600, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> kthread_run replaces kernel_thread and dameonize.
>
> allow_signal is unnecessary and has been removed.
> tid_poll was unused and has been removed.

Thread handling in this driver is quite interesting. Greg has his name
in there, so we can blame everything on him ;-)

Below is my take at cleaning at least the thread-related bits up:

- full switch to kthread infrastructure
- switch semOperations to a mutex, and give it a proper name
- remove the useless hpc_poll_thread wrapper
- remove ibmphp_hpc_initvars - everything left can easily be
  initialized statically

Signed-off-by: Christoph Hellwig <hch@lst.de>

Index: linux-2.6/drivers/pci/hotplug/ibmphp.h
==========
--- linux-2.6.orig/drivers/pci/hotplug/ibmphp.h 2007-04-22 12:44:04.000000000 +0200
+++ linux-2.6/drivers/pci/hotplug/ibmphp.h 2007-04-22 12:44:07.000000000 +0200
@@ -392,7 +392,6 @@ extern int ibmphp_add_pfmem_from_mem (st
 extern struct bus_node *ibmphp_find_res_bus (u8);
 extern void ibmphp_print_test (void); /* for debugging purposes */

-extern void ibmphp_hpc_initvars (void);
 extern int ibmphp_hpc_readslot (struct slot *, u8, u8 *);
 extern int ibmphp_hpc_writeslot (struct slot *, u8);
 extern void ibmphp_lock_operations (void);
Index: linux-2.6/drivers/pci/hotplug/ibmphp_core.c
==========
--- linux-2.6.orig/drivers/pci/hotplug/ibmphp_core.c 2007-04-22 12:44:11.000000000 +0200
+++ linux-2.6/drivers/pci/hotplug/ibmphp_core.c 2007-04-22 12:44:16.000000000 +0200
@@ -1368,8 +1368,6 @@ static int __init ibmphp_init(void)

  ibmphp_debug = debug;

- ibmphp_hpc_initvars();
-

```
  for (i = 0; i < 16; i++)
   irqs[i] = 0;

Index: linux-2.6/drivers/pci/hotplug/ibmphp_hpc.c
===================================================================
--- linux-2.6.orig/drivers/pci/hotplug/ibmphp_hpc.c 2007-04-22 12:31:23.000000000 +0200
+++ linux-2.6/drivers/pci/hotplug/ibmphp_hpc.c 2007-04-22 12:45:51.000000000 +0200
@@ -35,6 +35,7 @@
 #include <linux/smp_lock.h>
 #include <linux/init.h>
 #include <linux/mutex.h>
+#include <linux/kthread.h>

 #include "ibmphp.h"

@@ -101,12 +102,10 @@ static int to_debug = 0;
 //--------------------------------------------------------------------------
 // global variables
 //--------------------------------------------------------------------------
-static int ibmphp_shutdown;
-static int tid_poll;
-static struct mutex sem_hpcaccess; // lock access to HPC
-static struct semaphore semOperations; // lock all operations and
+static struct task_struct *ibmphp_poll_thread;
+static DEFINE_MUTEX(sem_hpcaccess); // lock access to HPC
+static DEFINE_MUTEX(ibmphp_op_sem); // lock all operations and
      // access to data structures
-static struct semaphore sem_exit; // make sure polling thread goes away
 //--------------------------------------------------------------------------
 // local function prototypes
 //--------------------------------------------------------------------------
@@ -116,33 +115,11 @@ static u8 hpc_writecmdtoindex (u8, u8);
 static u8 hpc_readcmdtoindex (u8, u8);
 static void get_hpc_access (void);
 static void free_hpc_access (void);
-static void poll_hpc (void);
 static int process_changeinstatus (struct slot *, struct slot *);
 static int process_changeinlatch (u8, u8, struct controller *);
-static int hpc_poll_thread (void *);
 static int hpc_wait_ctlr_notworking (int, struct controller *, void __iomem *, u8 *);
 //--------------------------------------------------------------------------

-
-/*--------------------------------------------------------------------
-* Name:    ibmphp_hpc_initvars
-*
-* Action:  initialize semaphores and variables
-*--------------------------------------------------------------------*/
```

```
-void __init ibmphp_hpc_initvars (void)
-{
- debug ("%s - Entry\n", __FUNCTION__);
-
- mutex_init(&sem_hpcaccess);
- init_MUTEX (&semOperations);
- init_MUTEX_LOCKED (&sem_exit);
- to_debug = 0;
- ibmphp_shutdown = 0;
- tid_poll = 0;
-
- debug ("%s - Exit\n", __FUNCTION__);
-}
-
 /*---------------------------------------------------------------------
 * Name:    i2c_ctrl_read
 *
@@ -798,7 +775,7 @@ void free_hpc_access (void)
 *---------------------------------------------------------------------*/
 void ibmphp_lock_operations (void)
 {
- down (&semOperations);
+ mutex_lock(&ibmphp_op_sem);
  to_debug = 1;
 }

@@ -808,7 +785,7 @@ void ibmphp_lock_operations (void)
 void ibmphp_unlock_operations (void)
 {
  debug ("%s - Entry\n", __FUNCTION__);
- up (&semOperations);
+ mutex_unlock(&ibmphp_op_sem);
  to_debug = 0;
  debug ("%s - Exit\n", __FUNCTION__);
 }
@@ -819,7 +796,7 @@ void ibmphp_unlock_operations (void)
 #define POLL_LATCH_REGISTER 0
 #define POLL_SLOTS  1
 #define POLL_SLEEP  2
-static void poll_hpc (void)
+static int poll_hpc(void *data)
 {
  struct slot myslot;
  struct slot *pslot = NULL;
@@ -833,12 +810,9 @@ static void poll_hpc (void)

  debug ("%s - Entry\n", __FUNCTION__);
```

```
- while (!ibmphp_shutdown) {
-  if (ibmphp_shutdown)
-   break;
-
+ while (!kthread_should_stop()) {
   /* try to get the lock to do some kind of hardware access */
-  down (&semOperations);
+  mutex_lock(&ibmphp_op_sem);

   switch (poll_state) {
   case POLL_LATCH_REGISTER:
@@ -893,14 +867,13 @@ static void poll_hpc (void)
    break;
   case POLL_SLEEP:
    /* don't sleep with a lock on the hardware */
-   up (&semOperations);
+   mutex_unlock(&ibmphp_op_sem);
    msleep(POLL_INTERVAL_SEC * 1000);

-   if (ibmphp_shutdown)
+   if (kthread_should_stop())
     break;

-   down (&semOperations);
-
+   mutex_lock(&ibmphp_op_sem);
    if (poll_count >= POLL_LATCH_CNT) {
     poll_count = 0;
     poll_state = POLL_SLOTS;
@@ -909,12 +882,13 @@ static void poll_hpc (void)
    break;
   }
   /* give up the hardware semaphore */
-  up (&semOperations);
+  mutex_unlock(&ibmphp_op_sem);
   /* sleep for a short time just for good measure */
   msleep(100);
  }
- up (&sem_exit);
  debug ("%s - Exit\n", __FUNCTION__);
+
+ return 0;
 }


@@ -1050,47 +1024,19 @@ static int process_changeinlatch (u8 old
 }
```

```
 /*-----------------------------------------------------------------------
-* Name:    hpc_poll_thread
-*
-* Action:  polling
-*
-* Return   0
-* Value:
-*-----------------------------------------------------------------------*/
-static int hpc_poll_thread (void *data)
-{
- debug ("%s - Entry\n", __FUNCTION__);
-
- daemonize("hpc_poll");
- allow_signal(SIGKILL);
-
- poll_hpc ();
-
- tid_poll = 0;
- debug ("%s - Exit\n", __FUNCTION__);
- return 0;
-}
-
-
-/*-----------------------------------------------------------------------
 * Name:    ibmphp_hpc_start_poll_thread
 *
 * Action:  start polling thread
 *-----------------------------------------------------------------------*/
 int __init ibmphp_hpc_start_poll_thread (void)
 {
- int rc = 0;
-
- debug ("%s - Entry\n", __FUNCTION__);
-
- tid_poll = kernel_thread (hpc_poll_thread, NULL, 0);
- if (tid_poll < 0) {
+ ibmphp_poll_thread = kthread_run(poll_hpc, NULL, "hpc_poll");
+ if (IS_ERR(ibmphp_poll_thread)) {
   err ("%s - Error, thread not started\n", __FUNCTION__);
-  rc = -1;
+  return PTR_ERR(ibmphp_poll_thread);
  }

- debug ("%s - Exit tid_poll[%d] rc[%d]\n", __FUNCTION__, tid_poll, rc);
- return rc;
+ return 0;
 }
```

```
 /*----------------------------------------------------------------------
@@ -1100,28 +1046,11 @@ int __init ibmphp_hpc_start_poll_thread
 *----------------------------------------------------------------------*/
 void __exit ibmphp_hpc_stop_poll_thread (void)
 {
- debug ("%s - Entry\n", __FUNCTION__);
+ kthread_stop(ibmphp_poll_thread);

- ibmphp_shutdown = 1;
- debug ("before locking operations \n");
  ibmphp_lock_operations ();
- debug ("after locking operations \n");
-
- // wait for poll thread to exit
- debug ("before sem_exit down \n");
- down (&sem_exit);
- debug ("after sem_exit down \n");
-
- // cleanup
- debug ("before free_hpc_access \n");
  free_hpc_access ();
- debug ("after free_hpc_access \n");
  ibmphp_unlock_operations ();
- debug ("after unlock operations \n");
- up (&sem_exit);
- debug ("after sem exit up\n");
-
- debug ("%s - Exit\n", __FUNCTION__);
 }

 /*----------------------------------------------------------------------
```

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