

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-09/msg09154.html>

- From: Henne <henne@xxxxxxxxxxxxxxxxxxxx>
 - Date: Sat, 30 Sep 2006 22:35:17 +0200
-

Changes the obsolete typedefd Scsi_Cmnd to struct scsi_cmnd in the ninja scsi pcmcia driver.

Signed-off-by: Henrik Kretzschmar <henne@xxxxxxxxxxxxxxxxxxxx>

```
nsp_cs.c | 42 ++++++-----
nsp_cs.h | 46 ++++++-----
nsp_debug.c | 4 ++--
nsp_message.c | 4 ++--
4 files changed, 51 insertions(+), 45 deletions(-)
```

```
diff -ruN linux-2.6/drivers/scsi/pcmcia/nsp_cs.c linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_cs.c
--- linux-2.6/drivers/scsi/pcmcia/nsp_cs.c 2006-08-01 01:31:43.000000000 +0200
+++ linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_cs.c 2006-09-30 22:08:48.000000000 +0200
@@ -184,7 +184,7 @@
```

```
* Clenaup parameters and call done() functions.
* You must be set SCpnt->result before call this function.
*/
```

```
-static void nsp_scsi_done(Scsi_Cmnd *SCpnt)
+static void nsp_scsi_done(struct scsi_cmnd *SCpnt)
{
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;
```

```
@@ -193,7 +193,8 @@
SCpnt->scsi_done(SCpnt);
}
```

```
-static int nsp_queuecommand(Scsi_Cmnd *SCpnt, void (*done)(Scsi_Cmnd *))
+static int nsp_queuecommand(struct scsi_cmnd *SCpnt,
+ void (*done)(struct scsi_cmnd *))
{
#ifdef NSP_DEBUG
/*unsigned int host_id = SCpnt->device->host->this_id;*/
@@ -366,7 +367,7 @@
/*
* Start selection phase
*/
-static int nsphw_start_selection(Scsi_Cmnd *SCpnt)
```

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

```
+static int nsp_hw_start_selection(struct scsi_cmnd *SCpnt)
{
unsigned int host_id = SCpnt->device->host->this_id;
unsigned int base = SCpnt->device->host->io_port;
@@ -447,7 +448,7 @@
/*
* setup synchronous data transfer mode
*/
-static int nsp_analyze_sdr(Scsi_Cmnd *SCpnt)
+static int nsp_analyze_sdr(struct scsi_cmnd *SCpnt)
{
unsigned char target = scmd_id(SCpnt);
// unsigned char lun = SCpnt->device->lun;
@@ -505,7 +506,7 @@
/*
* start ninja hardware timer
*/
-static void nsp_start_timer(Scsi_Cmnd *SCpnt, int time)
+static void nsp_start_timer(struct scsi_cmnd *SCpnt, int time)
{
unsigned int base = SCpnt->device->host->io_port;
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;
@@ -518,7 +519,8 @@
/*
* wait for bus phase change
*/
-static int nsp_negate_signal(Scsi_Cmnd *SCpnt, unsigned char mask, char *str)
+static int nsp_negate_signal(struct scsi_cmnd *SCpnt, unsigned char mask,
+ char *str)
{
unsigned int base = SCpnt->device->host->io_port;
unsigned char reg;
@@ -545,9 +547,9 @@
/*
* expect Ninja Irq
*/
-static int nsp_expect_signal(Scsi_Cmnd *SCpnt,
- unsigned char current_phase,
- unsigned char mask)
+static int nsp_expect_signal(struct scsi_cmnd *SCpnt,
+ unsigned char current_phase,
+ unsigned char mask)
{
unsigned int base = SCpnt->device->host->io_port;
int time_out;
@@ -580,7 +582,7 @@
/*
* transfer SCSI message
*/
-static int nsp_xfer(Scsi_Cmnd *SCpnt, int phase)
+static int nsp_xfer(struct scsi_cmnd *SCpnt, int phase)
```

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

```
{
unsigned int base = SCpnt->device->host->io_port;
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;
@@ -620,7 +622,7 @@
/*
 * get extra SCSI data from fifo
 */
-static int nsp_dataphase_bypass(Scsi_Cmnd *SCpnt)
+static int nsp_dataphase_bypass(struct scsi_cmnd *SCpnt)
{
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;
unsigned int count;
@@ -652,7 +654,7 @@
/*
 * accept reselection
 */
-static int nsp_reselected(Scsi_Cmnd *SCpnt)
+static int nsp_reselected(struct scsi_cmnd *SCpnt)
{
unsigned int base = SCpnt->device->host->io_port;
unsigned int host_id = SCpnt->device->host->this_id;
@@ -691,7 +693,7 @@
/*
 * count how many data transferd
 */
-static int nsp_fifo_count(Scsi_Cmnd *SCpnt)
+static int nsp_fifo_count(struct scsi_cmnd *SCpnt)
{
unsigned int base = SCpnt->device->host->io_port;
unsigned int count;
@@ -718,7 +720,7 @@
/*
 * read data in DATA IN phase
 */
-static void nsp_pio_read(Scsi_Cmnd *SCpnt)
+static void nsp_pio_read(struct scsi_cmnd *SCpnt)
{
unsigned int base = SCpnt->device->host->io_port;
unsigned long mmio_base = SCpnt->device->host->base;
@@ -813,7 +815,7 @@
/*
 * write data in DATA OUT phase
 */
-static void nsp_pio_write(Scsi_Cmnd *SCpnt)
+static void nsp_pio_write(struct scsi_cmnd *SCpnt)
{
unsigned int base = SCpnt->device->host->io_port;
unsigned long mmio_base = SCpnt->device->host->base;
@@ -906,7 +908,7 @@
/*
 * setup synchronous/asynchronous data transfer mode
```

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

```
*/
-static int nsp_nexus(Scsi_Cmnd *SCpnt)
+static int nsp_nexus(struct scsi_cmnd *SCpnt)
{
unsigned int base = SCpnt->device->host->io_port;
unsigned char target = scmd_id(SCpnt);
@@ -953,7 +955,7 @@
{
unsigned int base;
unsigned char irq_status, irq_phase, phase;
- Scsi_Cmnd *tmpSC;
+ struct scsi_cmnd *tmpSC;
unsigned char target, lun;
unsigned int *sync_neg;
int i, tmp;
@@ -1531,7 +1533,7 @@
/*-----*/

/*
-static int nsp_eh_abort(Scsi_Cmnd *SCpnt)
+static int nsp_eh_abort(struct scsi_cmnd *SCpnt)
{
nsp_dbg(NSP_DEBUG_BUSRESET, "SCpnt=0x%p", SCpnt);

@@ -1559,7 +1561,7 @@
return SUCCESS;
}

-static int nsp_eh_bus_reset(Scsi_Cmnd *SCpnt)
+static int nsp_eh_bus_reset(struct scsi_cmnd *SCpnt)
{
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;

@@ -1568,7 +1570,7 @@
return nsp_bus_reset(data);
}

-static int nsp_eh_host_reset(Scsi_Cmnd *SCpnt)
+static int nsp_eh_host_reset(struct scsi_cmnd *SCpnt)
{
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;

diff --ruN linux-2.6/drivers/scsi/pcmcia/nsp_cs.h linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_cs.h
--- linux-2.6/drivers/scsi/pcmcia/nsp_cs.h 2006-08-01 01:31:43.000000000 +0200
+++ linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_cs.h 2006-09-30 22:18:02.000000000 +0200
@@ -266,7 +266,7 @@

int TimerCount;
int SelectionTimeOut;
- Scsi_Cmnd *CurrentSC;
+ struct scsi_cmnd *CurrentSC;
```

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

```
//int CurrnetTarget;

int FifoCount;
@@ -319,30 +319,34 @@
int hostno,
#endif
int inout);
-static int nsp_queuecommand(Scsi_Cmnd *SCpnt, void (* done)(Scsi_Cmnd *SCpnt));
+static int nsp_queuecommand(struct scsi_cmnd *SCpnt,
+ void (* done)(struct scsi_cmnd *SCpnt));

/* Error handler */
-/*static int nsp_eh_abort (Scsi_Cmnd *SCpnt);*/
-/*static int nsp_eh_device_reset(Scsi_Cmnd *SCpnt);*/
-static int nsp_eh_bus_reset (Scsi_Cmnd *SCpnt);
-static int nsp_eh_host_reset (Scsi_Cmnd *SCpnt);
+/*static int nsp_eh_abort (struct scsi_cmnd *SCpnt);*/
+/*static int nsp_eh_device_reset(struct scsi_cmnd *SCpnt);*/
+static int nsp_eh_bus_reset (struct scsi_cmnd *SCpnt);
+static int nsp_eh_host_reset (struct scsi_cmnd *SCpnt);
static int nsp_bus_reset (nsp_hw_data *data);

/* */
static int nsphw_init (nsp_hw_data *data);
-static int nsphw_start_selection(Scsi_Cmnd *SCpnt);
-static void nsp_start_timer (Scsi_Cmnd *SCpnt, int time);
-static int nsp_fifo_count (Scsi_Cmnd *SCpnt);
-static void nsp_pio_read (Scsi_Cmnd *SCpnt);
-static void nsp_pio_write (Scsi_Cmnd *SCpnt);
-static int nsp_nexus (Scsi_Cmnd *SCpnt);
-static void nsp_scsi_done (Scsi_Cmnd *SCpnt);
-static int nsp_analyze_sdr (Scsi_Cmnd *SCpnt);
-static int nsp_negate_signal (Scsi_Cmnd *SCpnt, unsigned char mask, char *str);
-static int nsp_expect_signal (Scsi_Cmnd *SCpnt, unsigned char current_phase, unsigned char mask);
-static int nsp_xfer (Scsi_Cmnd *SCpnt, int phase);
-static int nsp_dataphase_bypass (Scsi_Cmnd *SCpnt);
-static int nsp_reselected (Scsi_Cmnd *SCpnt);
+static int nsphw_start_selection(struct scsi_cmnd *SCpnt);
+static void nsp_start_timer (struct scsi_cmnd *SCpnt, int time);
+static int nsp_fifo_count (struct scsi_cmnd *SCpnt);
+static void nsp_pio_read (struct scsi_cmnd *SCpnt);
+static void nsp_pio_write (struct scsi_cmnd *SCpnt);
+static int nsp_nexus (struct scsi_cmnd *SCpnt);
+static void nsp_scsi_done (struct scsi_cmnd *SCpnt);
+static int nsp_analyze_sdr (struct scsi_cmnd *SCpnt);
+static int nsp_negate_signal (struct scsi_cmnd *SCpnt,
+ unsigned char mask, char *str);
+static int nsp_expect_signal (struct scsi_cmnd *SCpnt,
+ unsigned char current_phase,
+ unsigned char mask);
+static int nsp_xfer (struct scsi_cmnd *SCpnt, int phase);
```

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

```
+static int nsp_dataphase_bypass (struct scsi_cmnd *SCpnt);
+static int nsp_reselected (struct scsi_cmnd *SCpnt);
static struct Scsi_Host *nsp_detect(struct scsi_host_template *sht);

/* Interrupt handler */
@@ -355,8 +359,8 @@

/* Debug */
#ifdef NSP_DEBUG
-static void show_command (Scsi_Cmnd *SCpnt);
-static void show_phase (Scsi_Cmnd *SCpnt);
+static void show_command (struct scsi_cmnd *SCpnt);
+static void show_phase (struct scsi_cmnd *SCpnt);
static void show_busphase(unsigned char stat);
static void show_message (nsp_hw_data *data);
#else
diff -ruN linux-2.6/drivers/scsi/pcmcia/nsp_debug.c linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_debug.c
--- linux-2.6/drivers/scsi/pcmcia/nsp_debug.c 2006-08-01 01:31:43.000000000 +0200
+++ linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_debug.c 2006-09-30 22:18:48.000000000 +0200
@@ -138,12 +138,12 @@
printk("\n");
}

-static void show_command(Scsi_Cmnd *SCpnt)
+static void show_command(struct scsi_cmnd *SCpnt)
{
print_commandk(SCpnt->cmnd);
}

-static void show_phase(Scsi_Cmnd *SCpnt)
+static void show_phase(struct scsi_cmnd *SCpnt)
{
int i = SCpnt->SCp.phase;

diff -ruN linux-2.6/drivers/scsi/pcmcia/nsp_message.c
linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_message.c
--- linux-2.6/drivers/scsi/pcmcia/nsp_message.c 2006-08-01 01:31:43.000000000 +0200
+++ linux-2.6.18-git14/drivers/scsi/pcmcia/nsp_message.c 2006-09-30 22:20:18.000000000 +0200
@@ -8,7 +8,7 @@

/* $Id: nsp_message.c,v 1.6 2003/07/26 14:21:09 elca Exp $ */

-static void nsp_message_in(Scsi_Cmnd *SCpnt)
+static void nsp_message_in(struct scsi_cmnd *SCpnt)
{
unsigned int base = SCpnt->device->host->io_port;
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;
@@ -50,7 +50,7 @@

}
```

[PATCH] scsi: convert ninja driver to struct scsi_cmnd

```
-static void nsp_message_out(Scsi_Cmnd *SCpnt)
+static void nsp_message_out(struct scsi_cmnd *SCpnt)
{
nsp_hw_data *data = (nsp_hw_data *)SCpnt->device->host->hostdata;
int ret = 1;
```

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

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