

[PATCH 2.6] Altix serial driver

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-06/5532.html>

From: Pat Gefre (pfg_at_sgi.com)

Date: 06/23/04

Date: Wed, 23 Jun 2004 09:48:43 -0500 (CDT)

To: akpm@osdl.org

I'm resending with the signed-off line.... sorry I forgot it.

2.6 patch for our console driver. We converted the driver to use the serial core functions. Also some changes to use sysfs/udev and a different major number.

-- Pat

Patrick Gefre

Silicon Graphics, Inc. (E-Mail) pfg@sgi.com

2750 Blue Water Rd (Voice) (651) 683-3127

Eagan, MN 55121-1400 (FAX) (651) 683-3054

Signed-off-by: Patrick Gefre <pfg@sgi.com>

```
# This is a BitKeeper generated diff -Nru style patch.
#
# ChangeSet
# 2004/06/18 17:06:40-05:00 erikj@attica.americas.sgi.com
# Makefile:
# Remove old sn_serial console driver from makefile
#
# drivers/char/Makefile
# 2004/06/18 17:06:14-05:00 erikj@attica.americas.sgi.com +0 -1
# Remove old sn_serial console driver from makefile
#
diff -Nru a/drivers/char/Makefile b/drivers/char/Makefile
--- a/drivers/char/Makefile 2004-06-18 17:10:28 -05:00
+++ b/drivers/char/Makefile 2004-06-18 17:10:28 -05:00
@@ -41,7 +41,6 @@
obj-$(CONFIG_RIO) += rio/ generic_serial.o
obj-$(CONFIG_HVC_CONSOLE) += hvc_console.o
obj-$(CONFIG_RAW_DRIVER) += raw.o
-obj-$(CONFIG_SGI_L1_SERIAL) += sn_serial.o
obj-$(CONFIG_VIOCONS) += viocons.o
obj-$(CONFIG_VIOTAPE) += viotape.o
```

Linux-Kernel: [PATCH 2.6] Altix serial driver

```
# This is a BitKeeper generated diff -Nru style patch.
#
# ChangeSet
# 2004/06/18 17:01:46-05:00 erikj@attica.americas.sgi.com
# Makefile:
# Adds new sn_console driver to the Makefile
# Kconfig:
# Adds new sn_console driver to Kconfig
# Remove old sn_serial driver from config
# sn_console.c:
# Implementation for new Altix console driver.
# .del-sn_serial.c~9c52f144ac55b7cb:
# Delete: drivers/char/sn_serial.c (removes old driver)
#
# drivers/serial/Makefile
# 2004/06/18 16:55:50-05:00 erikj@attica.americas.sgi.com +1 -0
# Adds new sn_console driver to the Makefile
#
# drivers/serial/Kconfig
# 2004/06/18 16:55:33-05:00 erikj@attica.americas.sgi.com +8 -1
# Adds new sn_console driver to Kconfig
#
# drivers/char/Kconfig
# 2004/06/18 16:54:35-05:00 erikj@attica.americas.sgi.com +0 -16
# Remove old sn_serial driver from config
#
# drivers/serial/sn_console.c
# 2004/06/18 16:32:27-05:00 erikj@attica.americas.sgi.com +1138 -0
#
# drivers/serial/sn_console.c
# 2004/06/18 16:32:27-05:00 erikj@attica.americas.sgi.com +0 -0
# BitKeeper file /data/lwork/attica3/erikj/linux-2.5-console/drivers/serial/sn_console.c
#
# BitK
+ * @s: String
+ * @len: Length
+ *
+ */
+static int
+snt_hw_puts_raw(const char *s, int len)
+{
+ /* this will call the PROM and not return until this is done */
+ return ia64_sn_console_putb(s, len);
+}
+
+/**
+ * snt_hw_puts_buffered - Send string to console, polled or interrupt mode
+ * @s: String
+ * @len: Length
+ */
```

Linux-Kernel: [PATCH 2.6] Altix serial driver

```
+ */
+static int
+snt_hw_puts_buffered(const char *s, int len)
+{
+ /* queue data to the PROM */
+ return ia64_sn_console_xmit_chars((char *)s, len);
+}
+
+/* uart interface structs
+ * These functions are associated with the uart_port that the serial core
+ * infrastructure calls.
+ *
+ * Note: Due to how the console works, many routines are no-ops.
+ */
+
+/**
+ * snp_type – What type of console are we?
+ * @port: Port to operate with (we ignore since we only have one port)
+ */
+static const char *
+snp_type(struct uart_port *port)
+{
+ return ("SGI SN L1");
+}
+
+/**
+ * snp_tx_empty – Is the transmitter empty? We pretend we're always empty
+ * @port: Port to operate on (we ignore since we only have one port)
+ */
+static unsigned int
+snp_tx_empty(struct uart_port *port)
+{
+ return 1;
+}
+
+/**
+ * snp_stop_tx – stop the transmitter – no-op for us
+ * @port: Port to operate on – we ignore – no-op function
+ * @tty_stop: Set to 1 if called via uart_stop
+ */
+static void
+snp_stop_tx(struct uart_port *port, unsigned int tty_stop)
+{
+}
+
+/**
+ * snp_release_port – Free i/o and resources for port – no-op for us
+ * @port: Port to operate on – we ignore – no-op function
```

Linux–Kernel: [PATCH 2.6] Altix serial driver

```
+ *
+ */
+static void
+snp_release_port(struct uart_port *port)
+{
+}
+
+/**
+ * snp_enable_ms – Force modem status interrupts on – no–op for us
+ * @port: Port to operate on – we ignore – no–op function
+ *
+ */
+static void
+snp_enable_ms(struct uart_port *port)
+{
+}
+
+/**
+ * snp_shutdown – shut down the port – free irq and disable – no–op for us
+ * @port: Port to shut down – we ignore
+ *
+ */
+static void
+snp_shutdown(struct uart_port *port)
+{
+}
+
+/**
+ * snp_set_mctrl – set control lines (dtr, rts, etc) – no–op for our console
+ * @port: Port to operate on – we ignore
+ * @mctrl: Lines to set/unset – we ignore
+ *
+ */
+static void
+snp_set_mctrl(struct uart_port *port, unsigned int mctrl)
+{
+}
+
+/**
+ * snp_get_mctrl – get control line info, we just return a static value
+ * @port: port to operate on – we only have one port so we ignore this
+ *
+ */
+static unsigned int
+snp_get_mctrl(struct uart_port *port)
+{
+ return TIOCM_CAR | TIOCM_RNG | TIOCM_DSR | TIOCM_CTS;
+}
+
+/**
+ * snp_stop_rx – Stop the receiver – we ignore this
```

Linux-Kernel: [PATCH 2.6] Altix serial driver

```
+ * @port: Port to operate on – we ignore
+ *
+ */
+static void
+snp_stop_rx(struct uart_port *port)
+{
+}
+
+/**
+ * snp_start_tx – Start transmitter
+ * @port: Port to operate on
+ * @tty_stop: Set to 1 if called via uart_start
+ *
+ */
+static void
+snp_start_tx(struct uart_port *port, unsigned int tty_stop)
+{
+ if (sal_console_port.sc_ops->sal_wakeup_transmit)
+ sal_console_port.sc_ops->sal_wakeup_transmit(&sal_console_port, TRANSMIT_BUFFERED);
+}
+
+/**
+ * snp_break_ctl – handle breaks – ignored by us
+ * @port: Port to operate on
+ * @break_state: Break state
+ *
+ */
+static void
+snp_break_ctl(struct uart_port *port, int break_state)
+{
+}
+
+/**
+ * snp_startup – Start up the serial port – always
```