

[RFC][PATCH] Way for platforms to alter built-in serial ports

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-09/9017.html>

From: Benjamin Herrenschmidt (*benh_at_kernel.crashing.org*)

Date: 09/30/04

To: Russell King <rmk@arm.linux.org.uk>

Date: Thu, 30 Sep 2004 18:50:48 +1000

Hi Russel !

I'm back with a finally useable fix for an oooooold problem, which is the way the serial table is supposed to be hard coded in `asm/serial.h` makes it really nasty to deal with for platforms like ppc where a given kernel can boot machines with different setups or even no built-in serial port at all.

(I currently have the 3 cases on ppc64: some normal ISA serial, some with different port/irq settings, and machines with no ports at all, all in the same kernel image).

The `early_serial_setup()` thing has never been practical to use, since it basically consist of "shoving" entries into the table, which is a bit ugly, requires knowledge of the table size if we want to remove all entries but N, that sort of thing, but the MAIN issue is that it's fundamentally incompatible with the driver beeing in a module.

What I propose is a way for the arch to provide it's own table along with the size of it via a function call. It's optional, based on a `#ifdef` defined by the arch in it's `asm/serial.h`. The only remaining tricky point is the fact that you used to size your static array of UART's based on the size of the table. So with my path, an arch that defines `ARCH_HAS_GET_LEGACY_SERIAL_PORTS` is supposed to provide both the new `get_legacy_serial_ports()` function, but also to define `UART_NR` to something sensible. I hope one day, we'll be able to convert 8250 to more dynamic allocation though.

With this patch, I fix all my problems of properly detecting built-in serial ports `_and_` not munging around with non existing ports on machines with no legacy HW with a single kernel image on ppc64 (and hopefully on ppc32 too). I have the ppc64-side of the patch beeing tested on various hardware at the moment and will submit it if you accept this one.

Regards,

Linux-Kernel: [RFC][PATCH] Way for platforms to alter built-in serial ports

Ben.

Signed-off-by: Benjamin Herrenschmidt <benh@kernel.crashing.org>

```
diff -urN linux-2.5/drivers/serial/8250.c linux-maple/drivers/serial/8250.c
--- linux-2.5/drivers/serial/8250.c 2004-09-30 18:31:42.560719874 +1000
+++ linux-maple/drivers/serial/8250.c 2004-09-30 16:27:39.421941590 +1000
@@ -112,11 +112,18 @@
#define SERIAL_PORT_DFNS
#endif

+#ifndef ARCH_HAS_GET_LEGACY_SERIAL_PORTS
static struct old_serial_port old_serial_port[] = {
    SERIAL_PORT_DFNS /* defined in asm/serial.h */
};
-
+static inline struct old_serial_port *get_legacy_serial_ports(unsigned int *count)
+{
+ *count = ARRAY_SIZE(old_serial_port);
+ return old_serial_port;
+}
#define UART_NR (ARRAY_SIZE(old_serial_port) + CONFIG_SERIAL_8250_NR_UARTS)
+#endif /* ARCH_HAS_GET_LEGACY_SERIAL_PORTS */
+

#ifdef CONFIG_SERIAL_8250_RSA

@@ -1839,22 +1846,28 @@
{
    struct uart_8250_port *up;
    static int first = 1;
+ struct old_serial_port *old_ports;
+ int count;
    int i;

    if (!first)
        return;
    first = 0;

- for (i = 0, up = serial8250_ports; i < ARRAY_SIZE(old_serial_port);
+ old_ports = get_legacy_serial_ports(&count);
+ if (old_ports == NULL)
+ return;
+
+ for (i = 0, up = serial8250_ports; i < count;
        i++, up++) {
- up->port.iobase = old_serial_port[i].port;
- up->port.irq = irq_canonicalize(old_serial_port[i].irq);
- up->port.uartclk = old_serial_port[i].baud_base * 16;
- up->port.flags = old_serial_port[i].flags;
- up->port.hub6 = old_serial_port[i].hub6;
```

Linux-Kernel: [RFC][PATCH] Way for platforms to alter built-in serial ports

```
- up->port.membase = old_serial_port[i].iomem_base;
- up->port.iotype = old_serial_port[i].io_type;
- up->port.regshift = old_serial_port[i].iomem_reg_shift;
+ up->port.iobase = old_ports[i].port;
+ up->port.irq = irq_canonicalize(old_ports[i].irq);
+ up->port.uartclk = old_ports[i].baud_base * 16;
+ up->port.flags = old_ports[i].flags;
+ up->port.hub6 = old_ports[i].hub6;
+ up->port.membase = old_ports[i].iomem_base;
+ up->port.iotype = old_ports[i].io_type;
+ up->port.regshift = old_ports[i].iomem_reg_shift;
      up->port.ops = &serial8250_ops;
      if (share_irqs)
          up->port.flags |= UPF_SHARE_IRQ;
diff -urN linux-2.5/drivers/serial/8250.h linux-maple/drivers/serial/8250.h
--- linux-2.5/drivers/serial/8250.h 2004-09-30 18:31:42.561719806 +1000
+++ linux-maple/drivers/serial/8250.h 2004-09-30 15:36:55.867448623 +1000
@@ -21,18 +21,6 @@
void serial8250_suspend_port(int line);
void serial8250_resume_port(int line);

-struct old_serial_port {
- unsigned int uart;
- unsigned int baud_base;
- unsigned int port;
- unsigned int irq;
- unsigned int flags;
- unsigned char hub6;
- unsigned char io_type;
- unsigned char *iomem_base;
- unsigned short iomem_reg_shift;
-};
-
/*
 * This replaces serial_uart_config in include/linux/serial.h
 */
diff -urN linux-2.5/include/linux/serial.h linux-maple/include/linux/serial.h
--- linux-2.5/include/linux/serial.h 2004-09-30 18:31:55.867785437 +1000
+++ linux-maple/include/linux/serial.h 2004-09-30 15:36:57.981697919 +1000
@@ -14,6 +14,21 @@
#include <asm/page.h>

/*
+ * Definition of a legacy serial port
+ */
+struct old_serial_port {
+ unsigned int uart;
+ unsigned int baud_base;
+ unsigned int port;
+ unsigned int irq;
+ unsigned int flags;
```

Linux-Kernel: [RFC][PATCH] Way for platforms to alter built-in serial ports

```
+ unsigned char hub6;
+ unsigned char io_type;
+ unsigned char *iomem_base;
+ unsigned short iomem_reg_shift;
+ };
+
+ /*
+  * Counters of the input lines (CTS, DSR, RI, CD) interrupts
+  */
```

—

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

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

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