[patch 11/14] sclp: simplify vt220 cleanup logic

Source: http://linux.derkeiler.com/Mailing−Lists/Kernel/2008−07/msg00282.html

- *From*: Martin Schwidefsky <schwidefsky@xxxxxxxxxx>
- *Date*: Tue, 01 Jul 2008 14:48:20 +0200

From: Peter Oberparleiter <peter.oberparleiter@xxxxxxxxxx>

Fix a number of sclp_vt220 cleanup problems:
* fix list_empty check after list_del()
* mark init−only flag as __initdata
* remove implicit dependency between slab_available() and num_pages
* straighten multiple init handling (use init count)

Signed−off−by: Peter Oberparleiter <peter.oberparleiter@xxxxxxxxxx>
Signed−off−by: Martin Schwidefsky <schwidefsky@xxxxxxxxxx>
−−−

drivers/s390/char/sclp_vt220.c | 55 ++++++++++++++++++−−−−−−−−−−−−−−−−−−−−
1 file changed, 28 insertions(+), 27 deletions(−)

Index: quilt−2.6/drivers/s390/char/sclp_vt220.c
==========
−−− quilt−2.6.orig/drivers/s390/char/sclp_vt220.c
+++ quilt−2.6/drivers/s390/char/sclp_vt220.c
@@ −82,8 +82,8 @@ static struct sclp_vt220_request *sclp_v
/* Number of characters in current request buffer */
static int sclp_vt220_buffered_chars;

−/* Flag indicating whether this driver has already been initialized */
−static int sclp_vt220_initialized = 0;
+/* Counter controlling core driver initialization. */
+static int __initdata sclp_vt220_init_count;

/* Flag indicating that sclp_vt220_current_request should really
* have been already queued but wasn't because the SCLP was processing
@@ −609,10 +609,8 @@ sclp_vt220_flush_buffer(struct tty_struc
sclp_vt220_emit_current();
}

−/*
− * Initialize all relevant components and register driver with system.
− */
−static void __init __sclp_vt220_cleanup(void)
+/* Release allocated pages. */

```
+static void __init __sclp_vt220_free_pages(void)
{
struct list_head *page, *p;

@@ −623,21 +621,30 @@ static void __init __sclp_vt220_cleanup(
else
free_bootmem((unsigned long) page, PAGE_SIZE);
}
− if (!list_empty(&sclp_vt220_register.list))
− sclp_unregister(&sclp_vt220_register);
− sclp_vt220_initialized = 0;
}

−static int __init __sclp_vt220_init(void)
+/* Release memory and unregister from sclp core. Controlled by init counting −
+ * only the last invoker will actually perform these actions. */
+static void __init __sclp_vt220_cleanup(void)
+{
+ sclp_vt220_init_count−−;
+ if (sclp_vt220_init_count != 0)
+ return;
+ sclp_unregister(&sclp_vt220_register);
+ __sclp_vt220_free_pages();
+}
+
+/* Allocate buffer pages and register with sclp core. Controlled by init
+ * counting − only the first invoker will actually perform these actions. */
+static int __init __sclp_vt220_init(int num_pages)
{
void *page;
int i;
− int num_pages;
int rc;

− if (sclp_vt220_initialized)
+ sclp_vt220_init_count++;
+ if (sclp_vt220_init_count != 1)
return 0;
− sclp_vt220_initialized = 1;
spin_lock_init(&sclp_vt220_lock);
INIT_LIST_HEAD(&sclp_vt220_empty);
INIT_LIST_HEAD(&sclp_vt220_outqueue);
@@ −649,24 +656,22 @@ static int __init __sclp_vt220_init(void
sclp_vt220_flush_later = 0;

/* Allocate pages for output buffering */
− num_pages = slab_is_available() ? MAX_KMEM_PAGES : MAX_CONSOLE_PAGES;
for (i = 0; i < num_pages; i++) {
if (slab_is_available())
page = (void *) get_zeroed_page(GFP_KERNEL | GFP_DMA);
else
```

```
page = alloc_bootmem_low_pages(PAGE_SIZE);
if (!page) {
− __sclp_vt220_cleanup();
− return −ENOMEM;
+ rc = −ENOMEM;
+ goto out;
}
list_add_tail((struct list_head *) page, &sclp_vt220_empty);
}
rc = sclp_register(&sclp_vt220_register);
+out:
if (rc) {
− printk(KERN_ERR SCLP_VT220_PRINT_HEADER
− "could not register vt220 − "
− "sclp_register returned %d\n", rc);
− __sclp_vt220_cleanup();
+ __sclp_vt220_free_pages();
+ sclp_vt220_init_count−−;
}
return rc;
}
@@ −689,15 +694,13 @@ static int __init sclp_vt220_tty_init(vo
{
struct tty_driver *driver;
int rc;
− int cleanup;

/* Note: we're not testing for CONSOLE_IS_SCLP here to preserve
* symmetry between VM and LPAR systems regarding ttyS1. */
driver = alloc_tty_driver(1);
if (!driver)
return −ENOMEM;
− cleanup = !sclp_vt220_initialized;
− rc = __sclp_vt220_init();
+ rc = __sclp_vt220_init(MAX_KMEM_PAGES);
if (rc)
goto out_driver;

@@ −723,8 +726,7 @@ static int __init sclp_vt220_tty_init(vo
return 0;

out_init:
− if (cleanup)
− __sclp_vt220_cleanup();
+ __sclp_vt220_cleanup();
out_driver:
put_tty_driver(driver);
return rc;
@@ −773,10 +775,9 @@ sclp_vt220_con_init(void)
{
int rc;
```

− INIT_LIST_HEAD(&sclp_vt220_register.list);
if (!CONSOLE_IS_SCLP)
return 0;
− rc = __sclp_vt220_init();
+ rc = __sclp_vt220_init(MAX_CONSOLE_PAGES);
if (rc)
return rc;
/* Attach linux console */

−−
blue skies,
Martin.

"Reality continues to ruin my life." − Calvin.

−−
To unsubscribe from this list: send the line "unsubscribe linux−kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxx
More majordomo info at http://vger.kernel.org/majordomo−info.html
Please read the FAQ at http://www.tux.org/lkml/