

---

Subject: [PATCH 11/12] proc: switch /proc/scsi/device\_info to seq\_file interface  
Posted by [Alexey Dobriyan](#) on Tue, 08 Apr 2008 15:03:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Note 1: 0644 should be used, but root bypasses permissions, so writing to /proc/scsi/device\_info still works.

Note 2: looks like scsi\_dev\_info\_list is unprotected

Note 3: probably make proc whine about "unwriteable but with ->write hook" entries. Probably.

Signed-off-by: Alexey Dobriyan <[adobriyan@sw.ru](mailto:adobriyan@sw.ru)>

---

drivers/scsi/scsi\_devinfo.c | 77 ++++++-----  
1 file changed, 44 insertions(+), 33 deletions(-)

--- a/drivers/scsi/scsi\_devinfo.c

+++ b/drivers/scsi/scsi\_devinfo.c

@@ -449,37 +449,40 @@ int scsi\_get\_device\_flags(struct scsi\_device \*sdev,  
{

#ifdef CONFIG\_SCSI\_PROC\_FS

/\*

- \* proc\_scsi\_dev\_info\_read: dump the scsi\_dev\_info\_list via

- \* /proc/scsi/device\_info

- \*/

-static int proc\_scsi\_devinfo\_read(char \*buffer, char \*\*start,

- off\_t offset, int length)

+static int devinfo\_seq\_show(struct seq\_file \*m, void \*v)

{

- struct scsi\_dev\_info\_list \*devinfo;

- int size, len = 0;

- off\_t begin = 0;

- off\_t pos = 0;

+ struct scsi\_dev\_info\_list \*devinfo =

+ list\_entry(v, struct scsi\_dev\_info\_list, dev\_info\_list);

- list\_for\_each\_entry(devinfo, &scsi\_dev\_info\_list, dev\_info\_list) {

- size = sprintf(buffer + len, "%0.8s' '%0.16s' 0x%x\n",

+ seq\_printf(m, "%0.8s' '%0.16s' 0x%x\n",

devinfo->vendor, devinfo->model, devinfo->flags);

- len += size;

- pos = begin + len;

- if (pos < offset) {

- len = 0;

- begin = pos;

- }

- if (pos > offset + length)

```

- goto stop_output;
- }
+ return 0;
+}
+
+static void * devinfo_seq_start(struct seq_file *m, loff_t *pos)
+{
+ return seq_list_start(&scsi_dev_info_list, *pos);
+}

-stop_output:
- *start = buffer + (offset - begin); /* Start of wanted data */
- len -= (offset - begin); /* Start slop */
- if (len > length)
- len = length; /* Ending slop */
- return (len);
+static void * devinfo_seq_next(struct seq_file *m, void *v, loff_t *pos)
+{
+ return seq_list_next(v, &scsi_dev_info_list, pos);
+}
+
+static void devinfo_seq_stop(struct seq_file *m, void *v)
+{
+}
+
+static const struct seq_operations scsi_devinfo_seq_ops = {
+ .start = devinfo_seq_start,
+ .next = devinfo_seq_next,
+ .stop = devinfo_seq_stop,
+ .show = devinfo_seq_show,
+};
+
+static int proc_scsi_devinfo_open(struct inode *inode, struct file *file)
+{
+ return seq_open(file, &scsi_devinfo_seq_ops);
+}

/*
@@ -489,11 +492,12 @@ stop_output:
 * integer value of flag to the scsi device info list.
 * To use, echo "vendor:model:flag" > /proc/scsi/device_info
 */
-static int proc_scsi_devinfo_write(struct file *file, const char __user *buf,
- unsigned long length, void *data)
+static ssize_t proc_scsi_devinfo_write(struct file *file,
+ const char __user *buf,
+ size_t length, loff_t *ppos)
{

```

```

    char *buffer;
- int err = length;
+ ssize_t err = length;

    if (!buf || length > PAGE_SIZE)
        return -EINVAL;
@@ -517,6 +521,15 @@ out:
    free_page((unsigned long)buffer);
    return err;
}
+
+static const struct file_operations scsi_devinfo_proc_fops = {
+ .owner = THIS_MODULE,
+ .open = proc_scsi_devinfo_open,
+ .read = seq_read,
+ .write = proc_scsi_devinfo_write,
+ .llseek = seq_lseek,
+ .release = seq_release,
+};
#endif /* CONFIG_SCSI_PROC_FS */

module_param_string(dev_flags, scsi_dev_flags, sizeof(scsi_dev_flags), 0);
@@ -577,15 +590,13 @@ int __init scsi_init_devinfo(void)
}

#ifdef CONFIG_SCSI_PROC_FS
- p = create_proc_entry("scsi/device_info", 0, NULL);
+ p = proc_create("scsi/device_info", 0, NULL, &scsi_devinfo_proc_fops);
    if (!p) {
        error = -ENOMEM;
        goto out;
    }

    p->owner = THIS_MODULE;
- p->get_info = proc_scsi_devinfo_read;
- p->write_proc = proc_scsi_devinfo_write;
#endif /* CONFIG_SCSI_PROC_FS */

out:
--
1.5.4.3

```

---