

[PATCH 2/2] ch: remove forward declarations

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-01/msg09586.html>

- *From:* FUJITA Tomonori <fujita.tomonori@xxxxxxxxxxxxxx>
 - *Date:* Thu, 24 Jan 2008 17:17:46 +0900
-

This moves ch_template and changer_fops structs to the end of file and removes forward declarations.

This also removes some trailing whitespace.

Signed-off-by: FUJITA Tomonori <fujita.tomonori@xxxxxxxxxxxxxx>

drivers/scsi/ch.c | 120 ++++++-----
1 files changed, 54 insertions(+), 66 deletions(-)

```
diff --git a/drivers/scsi/ch.c b/drivers/scsi/ch.c
index 2b07014..7aad154 100644
--- a/drivers/scsi/ch.c
+++ b/drivers/scsi/ch.c
@@ -90,16 +90,6 @@ static const char * vendor_labels[CH_TYPES-4] = {

#define MAX_RETRIES 1

-static int ch_probe(struct device *);
-static int ch_remove(struct device *);
-static int ch_open(struct inode * inode, struct file * filp);
-static int ch_release(struct inode * inode, struct file * filp);
-static long ch_ioctl(struct file *filp, unsigned int cmd, unsigned long arg);
-#ifdef CONFIG_COMPAT
-static long ch_ioctl_compat(struct file * filp,
- unsigned int cmd, unsigned long arg);
-#endif
-
static struct class * ch_sysfs_class;

typedef struct {
@@ -118,27 +108,6 @@ typedef struct {
static DEFINE_IDR(ch_index_idr);
static DEFINE_SPINLOCK(ch_index_lock);

-static struct scsi_driver ch_template =
- {
- .owner = THIS_MODULE,
- .gendrv = {
```

[PATCH 2/2] ch: remove forward declarations

```
- .name = "ch",
- .probe = ch_probe,
- .remove = ch_remove,
- },
-};
-
-static const struct file_operations changer_fops =
- {
- .owner = THIS_MODULE,
- .open = ch_open,
- .release = ch_release,
- .unlocked_ioctl = ch_ioctl,
-#ifdef CONFIG_COMPAT
- .compat_ioctl = ch_ioctl_compat,
-#endif
-};
-
static const struct {
unsigned char sense;
unsigned char asc;
@@ -207,7 +176,7 @@ ch_do_scsi(scsi_changer *ch, unsigned char *cmd,
{
int errno, retries = 0, timeout, result;
struct scsi_sense_hdr sshdr;
-
+
timeout = (cmd[0] == INITIALIZE_ELEMENT_STATUS)
? timeout_init : timeout_move;

@@ -245,7 +214,7 @@ static int
ch_elem_to_typecode(scsi_changer *ch, u_int elem)
{
int i;
-
+
for (i = 0; i < CH_TYPES; i++) {
if (elem >= ch->firsts[i] &&
elem < ch->firsts[i] +
@@ -261,15 +230,15 @@ ch_read_element_status(scsi_changer *ch, u_int elem, char *data)
u_char cmd[12];
u_char *buffer;
int result;
-
+
buffer = kmalloc(512, GFP_KERNEL | GFP_DMA);
if(!buffer)
return -ENOMEM;
-
+
retry:
memset(cmd,0,sizeof(cmd));
```

[PATCH 2/2] ch: remove forward declarations

```
cmd[0] = READ_ELEMENT_STATUS;
- cmd[1] = (ch->device->lun << 5) |
+ cmd[1] = (ch->device->lun << 5) |
(ch->voltags ? 0x10 : 0) |
ch_elem_to_typecode(ch,elem);
cmd[2] = (elem >> 8) & 0xff;
@@ -296,7 +265,7 @@ ch_read_element_status(scsi_changer *ch, u_int elem, char *data)
return result;
}
```

```
-static int
+static int
ch_init_elem(scsi_changer *ch)
{
int err;
@@ -322,7 +291,7 @@ ch_readconfig(scsi_changer *ch)
buffer = kzalloc(512, GFP_KERNEL | GFP_DMA);
if (!buffer)
return -ENOMEM;
-
+
memset(cmd,0,sizeof(cmd));
cmd[0] = MODE_SENSE;
cmd[1] = ch->device->lun << 5;
@@ -365,7 +334,7 @@ ch_readconfig(scsi_changer *ch)
} else {
vprintk("reading element address assignment page failed!\n");
}
-
+
/* vendor specific element types */
for (i = 0; i < 4; i++) {
if (0 == vendor_counts[i])
@@ -443,7 +412,7 @@ static int
ch_position(scsi_changer *ch, u_int trans, u_int elem, int rotate)
{
u_char cmd[10];
-
+
dprintk("position: 0x%x\n",elem);
if (0 == trans)
trans = ch->firsts[CHET_MT];
@@ -462,7 +431,7 @@ static int
ch_move(scsi_changer *ch, u_int trans, u_int src, u_int dest, int rotate)
{
u_char cmd[12];
-
+
dprintk("move: 0x%x => 0x%x\n",src,dest);
if (0 == trans)
trans = ch->firsts[CHET_MT];
```

[PATCH 2/2] ch: remove forward declarations

```
@@ -484,7 +453,7 @@ ch_exchange(scsi_changer *ch, u_int trans, u_int src,
u_int dest1, u_int dest2, int rotate1, int rotate2)
{
u_char cmd[12];
-
+
dprintk("exchange: 0x%x => 0x%x => 0x%x\n",
src,dest1,dest2);
if (0 == trans)
@@ -501,7 +470,7 @@ ch_exchange(scsi_changer *ch, u_int trans, u_int src,
cmd[8] = (dest2 >> 8) & 0xff;
cmd[9] = dest2 & 0xff;
cmd[10] = (rotate1 ? 1 : 0) | (rotate2 ? 2 : 0);
-
+
return ch_do_scsi(ch, cmd, NULL,0, DMA_NONE);
}
```

```
@@ -539,14 +508,14 @@ ch_set_voltag(scsi_changer *ch, u_int elem,
elem, tag);
memset(cmd,0,sizeof(cmd));
cmd[0] = SEND_VOLUME_TAG;
- cmd[1] = (ch->device->lun << 5) |
+ cmd[1] = (ch->device->lun << 5) |
ch_elem_to_typecode(ch,elem);
cmd[2] = (elem >> 8) & 0xff;
cmd[3] = elem & 0xff;
cmd[5] = clear
? (alternate ? 0x0d : 0x0c)
: (alternate ? 0x0b : 0x0a);
-
+
cmd[9] = 255;
```

```
memcpy(buffer,tag,32);
@@ -562,7 +531,7 @@ static int ch_gstatus(scsi_changer *ch, int type, unsigned char __user *dest)
int retval = 0;
u_char data[16];
unsigned int i;
-
+
mutex_lock(&ch->lock);
for (i = 0; i < ch->counts[type]; i++) {
if (0 != ch_read_element_status
@@ -629,18 +598,18 @@ static long ch_ioctl(struct file *file,
scsi_changer *ch = file->private_data;
int retval;
void __user *argp = (void __user *)arg;
-
+
switch (cmd) {
```

[PATCH 2/2] ch: remove forward declarations

[PATCH 2/2] ch: remove forward declarations

```
case CHIOGPARAMS:
{
struct changer_params params;
-
+
params.cp_curpicker = 0;
params.cp_npickers = ch->counts[CHET_MT];
params.cp_nslots = ch->counts[CHET_ST];
params.cp_nportals = ch->counts[CHET_IE];
params.cp_ndrives = ch->counts[CHET_DT];
-
+
if (copy_to_user(argp, &params, sizeof(params)))
return -EFAULT;
return 0;
@@ -670,11 +639,11 @@ static long ch_ioctl(struct file *file,
return -EFAULT;
return 0;
}
-
+
case CHIOPOSITION:
{
struct changer_position pos;
-
+
if (copy_from_user(&pos, argp, sizeof(pos)))
return -EFAULT;

@@ -689,7 +658,7 @@ static long ch_ioctl(struct file *file,
mutex_unlock(&ch->lock);
return retval;
}
-
+
case CHIOMOVE:
{
struct changer_move mv;
@@ -702,7 +671,7 @@ static long ch_ioctl(struct file *file,
dprintk("CHIOMOVE: invalid parameter\n");
return -EBADSLT;
}
-
+
mutex_lock(&ch->lock);
retval = ch_move(ch,0,
ch->firsts[mv.cm_fromtype] + mv.cm_fromunit,
@@ -715,7 +684,7 @@ static long ch_ioctl(struct file *file,
case CHIOEXCHANGE:
{
struct changer_exchange mv;
```

[PATCH 2/2] ch: remove forward declarations

```
-
+
if (copy_from_user(&mv, argp, sizeof (mv)))
return -EFAULT;

@@ -725,7 +694,7 @@ static long ch_ioctl(struct file *file,
dprintk("CHIOEXCHANGE: invalid parameter\n");
return -EBADSLT;
}
-
+
mutex_lock(&ch->lock);
retval = ch_exchange
(ch,0,
@@ -740,7 +709,7 @@ static long ch_ioctl(struct file *file,
case CHIOGSTATUS:
{
struct changer_element_status ces;
-
+
if (copy_from_user(&ces, argp, sizeof (ces)))
return -EFAULT;
if (ces.ces_type < 0 || ces.ces_type >= CH_TYPES)
@@ -756,19 +725,19 @@ static long ch_ioctl(struct file *file,
u_char *buffer;
unsigned int elem;
int result,i;
-
+
if (copy_from_user(&cge, argp, sizeof (cge)))
return -EFAULT;

if (0 != ch_checkrange(ch, cge.cge_type, cge.cge_unit))
return -EINVAL;
elem = ch->firsts[cge.cge_type] + cge.cge_unit;
-
+
buffer = kmalloc(512, GFP_KERNEL | GFP_DMA);
if (!buffer)
return -ENOMEM;
mutex_lock(&ch->lock);
-
+
voltag_retry:
memset(cmd,0,sizeof(cmd));
cmd[0] = READ_ELEMENT_STATUS;
@@ -779,7 +748,7 @@ static long ch_ioctl(struct file *file,
cmd[3] = elem & 0xff;
cmd[5] = 1;
cmd[9] = 255;
-
```

[PATCH 2/2] ch: remove forward declarations

```
+
if (0 == (result = ch_do_scsi(ch, cmd, buffer, 256, DMA_FROM_DEVICE))) {
cge.cge_status = buffer[18];
cge.cge_flags = 0;
@@ -819,7 +788,7 @@ static long ch_ioctl(struct file *file,
}
kfree(buffer);
mutex_unlock(&ch->lock);
-
+
if (copy_to_user(argp, &cge, sizeof (cge)))
return -EFAULT;
return result;
@@ -832,7 +801,7 @@ static long ch_ioctl(struct file *file,
mutex_unlock(&ch->lock);
return retval;
}
-
+
case CHIOSVOLTAG:
{
struct changer_set_voltag csv;
@@ -873,7 +842,7 @@ static long ch_ioctl_compat(struct file * file,
unsigned int cmd, unsigned long arg)
{
scsi_changer *ch = file->private_data;
-
+
switch (cmd) {
case CHIOGPARAMS:
case CHIOGVPARAMS:
@@ -889,7 +858,7 @@ static long ch_ioctl_compat(struct file * file,
{
struct changer_element_status32 ces32;
unsigned char __user *data;
-
+
if (copy_from_user(&ces32, (void __user *)arg, sizeof (ces32)))
return -EFAULT;
if (ces32.ces_type < 0 || ces32.ces_type >= CH_TYPES)
@@ -982,10 +951,29 @@ static int ch_remove(struct device *dev)
return 0;
}

+static struct scsi_driver ch_template = {
+ .owner = THIS_MODULE,
+ .gendrv = {
+ .name = "ch",
+ .probe = ch_probe,
+ .remove = ch_remove,
+ },

```

[PATCH 2/2] ch: remove forward declarations

```
+};
+
+static const struct file_operations changer_fops = {
+.owner = THIS_MODULE,
+.open = ch_open,
+.release = ch_release,
+.unlocked_ioctl = ch_ioctl,
+#ifdef CONFIG_COMPAT
+.compat_ioctl = ch_ioctl_compat,
+#endif
+};
+
+static int __init init_ch_module(void)
+{
+int rc;
+
+
+printk(KERN_INFO "SCSI Media Changer driver v" VERSION "\n");
+ch_sysfs_class = class_create(THIS_MODULE, "scsi_changer");
+if (IS_ERR(ch_sysfs_class)) {
+@@ -1010,7 +998,7 @@ static int __init init_ch_module(void)
+return rc;
+}

-static void __exit exit_ch_module(void)
+static void __exit exit_ch_module(void)
+{
+scsi_unregister_driver(&ch_template.gendrv);
+unregister_chrdev(SCSI_CHANGER_MAJOR, "ch");
+--
1.5.3.4
--
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at http://vger.kernel.org/majordomo-info.html
Please read the FAQ at http://www.tux.org/lkml/
```