

Re: Problem with tcgetattr

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.system/2003-09/0163.html>

From: Peter T. Breuer (*ptb_at_oboe.it.uc3m.es*)

Date: 09/06/03

Date: Sat, 6 Sep 2003 13:33:55 +0200

Laurent H. <l.huge@nospam.wanadoo.fr> wrote:

> *Peter T. Breuer a 'ecrit :*

>> *That makes no difference. Which module registers the char device?*

> *I'm not. I've only implemented a line discipline for the serial port,
> but all the rest is the serial driver.*

>> *That's the one whose ioctl functions are used.*

>>> *\$ strace stty < /dev/ttyS0*

>>> *...*

>>> *ioctl(0, SNDCTL_TMR_TIMEBASE, 0xbfff7f0) = -1 EINVAL*

>>> *(Invalid argument)*

>> *There you are. You are intercepting the ioctl or else replacing the*

>> *serial drivers ioctl. And not doing it right.*

> *I've got not ioctl in my module, so what's doing it ?*

That's irrelevant. The person who has registered for the major/minor is "doing it". You have to register a struct of file ops, which contains various function addresses. These are the methods of the driver object. They will be called for write, read, ioctl, open, close, etc. If you didn't register one, and nobody else did, nobody is "doing it", and you will get -EINVAL or an oops.

> *\$ cat /proc/devices*

> *Character devices:*

> *1 mem*

> *2 pty*

> *3 tty*

> *4 ttyS*

Well, it looks like the serial driver to me. So you must be doing something to the registered methods.

> *The char device ccsds is a simple rw device in which I derivate the*

> *CCSDS flow (spatial standard) ; it has nothing to do with serial*

> *port, and I'm sure it doesn't interact with it.*

> *So the only thing I do against serial module is the line discipline :*

Uhhhhhhhhhh.

```
> struct tty_ldisc sport_ldisc = {
> magic : TTY_LDISC_MAGIC,
> name : "ccsds_sport_ldisc",
> flags : 0,
> open : sport_open,
> close : sport_close,
> receive_buf : sport_receive,
> receive_room : sport_room,
> };
```

```
> which I register with tty_register_ldisc(N_TTY, &sport_ldisc) during
> the module initialization.
> Does that mean I replace the serial duty on ioctl ?
```

It means whatever it means. To answer that question I would have to go and read the code! I'll leave it to you to do that.

Uhm. The struct is defined in tty_ldisc.h. And yes, you need an ioctl function.

```
* int (*ioctl)(struct tty_struct * tty, struct file * file,
* unsigned int cmd, unsigned long arg);
*
* This function is called when the user requests an ioctl which
* is not handled by the tty layer or the low-level tty driver.
* It is intended for ioctls which affect line discipline
* operation. Note that the search order for ioctls is (1) tty
* layer, (2) tty low-level driver, (3) line discipline. So a
* low-level driver can "grab" an ioctl request before the line
* discipline has a chance to see it.
```

and it's a field of your struct that you forgot to initialize and which might be any old random thing ...

```
struct tty_ldisc {
    int magic;
    char *name;
    int num;
    int flags;
    /*
    * The following routines are called from above.
    */
    int (*open)(struct tty_struct *);
    void (*close)(struct tty_struct *);
    void (*flush_buffer)(struct tty_struct *tty);
    ssize_t (*chars_in_buffer)(struct tty_struct *tty);
    ssize_t (*read)(struct tty_struct * tty, struct file * file,
        unsigned char * buf, size_t nr);
    ssize_t (*write)(struct tty_struct * tty, struct file * file,
```

```
    const unsigned char * buf, size_t nr);
int (*ioctl)(struct tty_struct * tty, struct file * file,
    ^^^^^^^^^
    unsigned int cmd, unsigned long arg);
void (*set_termios)(struct tty_struct *tty, struct termios * old);
unsigned int (*poll)(struct tty_struct *, struct file *,
    struct poll_table_struct *);

/*
 * The following routines are called from below.
 */
void (*receive_buf)(struct tty_struct *, const unsigned char *cp,
    char *fp, int count);
int (*receive_room)(struct tty_struct *);
void (*write_wakeup)(struct tty_struct *);
};
```

Peter