The HyperNews Linux KHG Discussion Pages

Device Drivers

If you choose to write a device driver, you must take everything written here as a guide, and no more. I cannot guarantee that this chapter will be free of errors, and I cannot guarantee that you will not damage your computer, even if you follow these instructions exactly. It is highly unlikely that you will damage it, but I cannot guarantee against it. There is only one ``infallible'' direction I can give you: **Back up!** Back up before you test your new device driver, or you may regret it later.

What is a Device Driver?
: What is this ``device driver'' stuff anyway? Here's a **very** short introduction to the concept.

User-space device drivers
: It's not always necessary to write a ``real'' device driver. Sometimes you just need to know how to write code that runs as a normal user process and still accesses hardware.

Device Driver Basics
: Assuming that you need to write a ``real'' device driver, there are some things that you need to know regardless of what type of driver you are writing. In fact, you may need to learn what type of driver you ought to write...

Character Device Drivers
: This section includes details specific to character device drivers, and assumes that you know everything in the previous section.

TTY drivers
: This section hasn't been written yet. TTY drivers are character devices that interface with the kernel's generic TTY support, and they require more than just a standard character device interface. I'd appreciate it if someone would write up how to attach a character device driver to the generic TTY layer and submit it to me for inclusion in this guide.

Block Device Drivers
: This section includes details specific to block device drivers (suprise!)

Writing a SCSI Device Driver
: This is a technical paper written by Rik Faith at the University of North Carolina.

Network Device Drivers
: Alan Cox gives an introduction to the network layer, including device drivers.

Supporting Functions
: Many functions are useful to all sorts of drivers. Here is a summary of quite a few of them.

Translating Addresses in Kernel Space
: An edited version of a post of Linus Torvalds to the linux-kernel mailing list about how to correctly deal with translating memory references when writing kernel source code such as device drivers.

Kernel-Level Exception Handling
: An edited version of a post of Joerg Pommnitz to the linux-kernel mailing list about how the new (Linux 2.1.8) exception mechanism works.

Other sources of information

Quite a few other references are also available on the topic of writing Linux device drivers by now. I put up some (slightly outdated by now, but still worth reading, I think) notes for a talk I gave in May 1995 entitled Writing Linux Device Drivers, which is specifically oriented at character devices implemented as kernel runtime-loadable modules.

Linux Journal has had a long-running series of articles called **Kernel Korner** which, despite the wacky name, has had quite a bit of useful information on it. Some of the articles from that column may be available on the web; most of them are available for purchase as back issues. One particularly useful series of articles, which focussed in far more detail than my 30 minute talk on the subject of kernel runtime-loadable modules, was in issues 23, 24, 25, 26, and 28. They were written by Alessandro Rubini and Georg v. Zezschwitz. Issue 29 is slated (as of this writing) to have an article on writing network device drivers, written by Alan Cox. Issues 9, 10, and 11 have a series that I wrote on block device drivers.

Messages

22. ❓ DMA to user space *by Marcel Boosten* NEW

21. How a device driver can driver his device *by Kim yeonseop* NEW

    1. Untitled NEW

20. memcpy error? *by Edgar Vonk*
19. Unable to handle kernel paging request - error *by Edgar Vonk*
17. _syscallX() Macros *by Tom Howley*
16. MediaMagic Sound Card DSP-16. How to run in Linux. *by Robert Hinson*
15. What does mark_bh() do? *by Erik Petersen*

    1. Untitled *by Praveen Dwivedi*

14. 3D Acceleration *by jamesbat@innotts.co.uk*
13. Device Drivers: /dev/radio... *by Matthew Kirkwood*
12. Does anybody know why kernel wakes my driver up without apparant reasons? *by David van Leeuwen*
11. Getting a DMA buffer aligned with 64k boundaries *by Juan de La Figuera Bayon*
10. Hardware Interface I/O Access *by Terry Moore*

    1. You are somewhat confused... *by Michael K. Johnson*

9. Is Anybody know something about SIS 496 IDE chipset? *by Alexander*
7. Vertical Retrace Interrupt - I need to use it *by Brynn Rogers*

    1. Your choice... *by Michael K. Johnson*

6. help working with skb structures *by arkane*
5. Interrupt Sharing ? *by Frieder Löffler*

    1. Interrupt sharing-possible *by Vladimir Myslik*
    -> Interrupt sharing - How to do with Network Drivers? *by Frieder Löffler*
    -> Interrupt sharing 101 *by Christophe Beauregard*

4. Device Driver notification of "Linux going down" *by Stan Troeh*

    1. Through application which has opened the device *by Michael K. Johnson*
    2. Device Driver notification of "Linux going down" *by Marko Kohtala*

3. Is waitv honored? *by Michael K. Johnson*
2. PCI Driver *by Flavia Donno*

    1. There is linux-2.0/drivers/pci/pci.c *by Hasdi*

1. Re: Network Device Drivers *by Paul Gortmaker*

    1. Re: Network Device Drivers *by Neal Tucker*

        1. network driver info *by Neal Tucker*
        -> Network Driver Desprately Needed *by Paul Atkinson*

    2. Transmit function *by Joerg Schorr*

        1. Re: Transmit function *by Paul Gortmaker*
        -> Skbuff *by Joerg Schorr*