

Linux Ethernet device driver demystified

- for Linux version 3.5.0

By Jie Yan

yanjie111@gmail.com

2015

1 Introduction

There are several network device driver tutorials of Linux on the internet. Why do I want to write another tutorial? There are 2 reasons: At first, almost all the other tutorials are talking about Linux version 2.6.x. There are a lot of code changes since then and it is hard for readers to study it by referencing current code. For example, in the 2.6.x version the transmission of packet calls the function inside the structure `net_device->hard_start_xmit()`, now it calls `net_device->netdev_ops->ndo_start_xmit()`. Secondly, because the network driver is device-specific, if the writer use memory-based driver module the readers can not understand how the real hardware works. Some tutorials used specified hardware, but the readers who have not the same hardware can not test the sample code.

This article will help the reader to understand and develop a network driver for their computers. The hardware specified codes are marked and I will tell the readers how to get the hardware specified information and write them to the code. The readers can do it step by step and test it on their computers with confidence.

Here is the overview of this article. Chapter 2 helps you to set up the build environments. You can build Linux device driver module for your computer. Chapter 3 lets you write a network driver that initialize the hardware. Chapter 4 can make your driver to transmit packets and get statistics. Chapter 5 enables your driver to receive packets. The driver is minimal; you can test it on your computer. After you have understood all the steps, it is easy for you to read the source code of the professional-grade driver.

2 Preparing for Driver Development

The hardware you need is a computer with internet connection.

The software you need is Linux installed on your PC. Either Ubuntu or Fedora is OK. This code was tested in Ubuntu 12.10 on an Acer Laptop PC.

2 things will be done in this chapter:

1. Get the Linux source code of your current version.
2. Get the compiler tools.

2.1 Get the Linux source code

At first you need to know your current Ubuntu Linux version of your PC.

```
$uname -r  
3.5.0-51-generic
```

Only the first 3 numbers are necessary. Delete the “-51-generic” of the output string and use the output to replace the “3.5.0” in the next command.

```
$sudo apt-get install build-essential linux-source-3.5.0 linux-headers-generic
```

After that command, you get the compressed tar file of the Linux source codes of your current version. They are located at /usr/src/ directory.

To list the files, run

```
$ls -l /usr/src
drwxr-xr-x  4 root root 4096 Jan 12 12:24 linux-source-3.5.0/
lrwxrwxrwx  1 root root  45 May 15  2014 linux-source-3.5.0.tar.bz2 ->
                                         linux-source-3.5.0/linux-source-3.5.0.tar.bz2
```

Change to the directory, extract the tar ball and create a link.

```
$cd /usr/src
$sudo tar -xvjf linux-source-3.5.0.tar.bz2
$sudo ln -s /usr/src/linux-source-3.5.0 /usr/src/linux
```

Now you have all the source codes in the directory /usr/src/linux-source-3.5.0 and a link to it.

Question 1: Why do I need the source code of current Linux version the same as my PC? Why not use the latest Linux source?

Answer: To make sure the device driver you build later can run on your PC.

Question 2: Why do I need a link to the current source code?

Answer: It is convenient to be used for building your device driver.

2.2 Get the compiler tools