

# 第一章：设备驱动程序简介

前言：

这一章主要介绍了以下几个部分：

1. 设备驱动程序的作用
2. 内核功能划分
3. 可装载模块
4. 设备和模块的分类
5. 安全问题
6. 版本编号
7. 许可证条款(GPL)

接下来就一一总结这 7 个部分的内容：

## 1. 设备驱动程序的作用

这一节主要介绍了一个核心的思想，就是，**设备驱动程序的作用在于提供机制(Machanism)，而不是提供策略(Policy)**

编程的问题都可以划分为 2 个部分：

- a) 需要提供什么功能(机制)
- b) 如何使用这些功能(策略)

**注意：**编写驱动程序时，程序员应该特别注意：编写访问硬件的内核代码时，不要给用户强加任何特定的策略，驱动程序应该处理如何使硬件可用的问题，而将怎样使用硬件的问题留给上层应用程序。这样的驱动程序才会比较灵活。

## 2. 内核功能的划分

内核功能分为以下几个部分：

- a) 进程管理  
负责创建和销毁进程，处理它们和外部世界之间的联系。
- b) 内存管理  
内核在有限的内存资源上为每个进程创建了一个虚拟的地址空间。
- c) 文件系统  
Linux 中，几乎每个对象都可以当做文件来处理，In linux, everything can be treated as a file.

d) 设备控制

几乎每一个系统操作最终都会映射到物理设备上。除了处理器，内存以及其它很有限的几个对象外。所以设备控制的操作都由被控制设备相关的代码来完成，这些代码就是驱动程序。

e) 网络功能

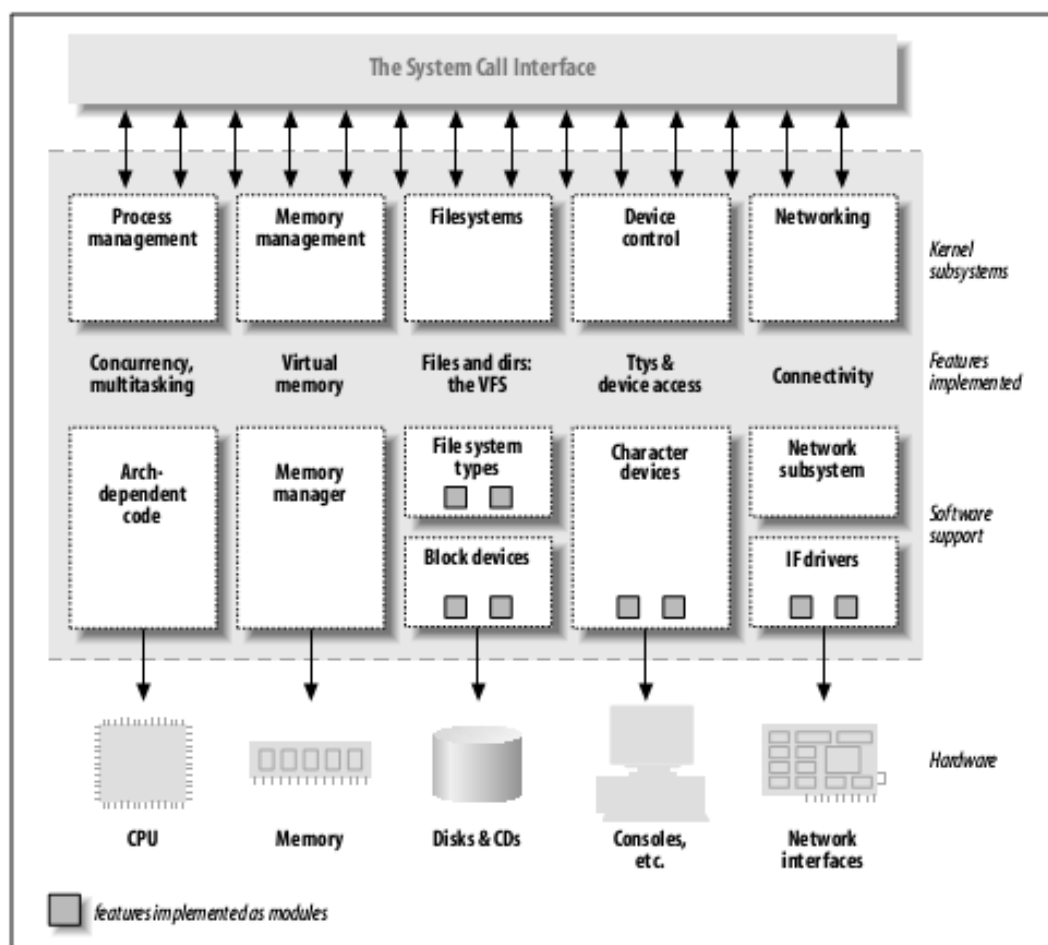
网络功能由操作系统来管理。大部分网络操作和具体进程无关，因为网络接收发送是异步事件。

### 3. 可装载模块

Linux 提供了一种很好的机制，就是在运行时可以装载(insmod)或移除(rmmod)驱动模块。

### 4. 设备和模块的分类

Linux 系统将设备分成三种基本类型，分别为：字符设备，块设备和网络设备。



a) 字符设备

字符(char)设备是个能像**字节流(类似文件)**一样被访问的设备，由字符设备

驱动程序来实现这种特性。

字符设备驱动至少要实现 `open`, `read`, `write`, `close` 系统调用。

字符设备可以通过文件系统节点来访问。字符设备文件和普通文件的区别在于，普通文件可以前后移动访问的位置，而字符设备文件只是一个可以顺序访问的数据通道。字符设备节点的例子：`/dev/console`, `/dev/ttyS0` 等。

#### b) 块设备

块设备也是通过访问 `/dev` 目录下的文件节点，比如 `/dev/block3` 或 `/dev/mtdblock3` 等。块设备在进行 I/O 操作时每次只能传输一个或多个完整的块，每块的大小为 512 字节(或 2 的更高次幂字节的数据)。**块设备与字符设备的区别仅仅在于内核内部管理数据的方式，也就是内核及驱动程序之间的软件接口，这些区别对用户来讲是透明的。**

#### c) 网络设备

任何网络处理都通过一个网络接口，即一个能够与其他主机交换数据的设备。通常接口是一个硬件设备，但也可能是一个纯软件设备，比如回环(loopback)接口。

另一种分类：某些驱动程序类型同内核用来支持某种给定类型设备的附加层一起工作。比如串行总线(USB)模块、串行模块、SCSI 模块，等等。每个 USB 设备由一个 USB 模块驱动，而该 USB 模块和 USB 子系统一同工作，但设备本身在系统中表现为一个字符设备(USB 串口)，一个块设备(U 盘)，或者一个网络设备(USB 以太网接口)。

## 5. 安全问题

### 原则：

- a) 驱动程序编写者应当尽量避免在代码中实现安全策略。安全策略最好在系统管理员的控制之下，由内核的高层来实现。
- b) 驱动程序编写者还应当避免由于自身原因引入的安全方面的缺陷。
  - c 编程语言很容易出现的几种类型的错误
    - i) 缓冲区溢出会导致很多安全问题
    - ii) 小心对待未初始化的内存
- c) 小心使用从第三方获得的软件。

## 6. 版本编号

此书基于 linux-2.6.10

## 7. 许可证条款

许可证的主要目的是通过允许每个人自由修改程序来实现知识增长；同时，向公众出售软件的人依旧可以获利。