<https://blog.csdn.net/m0_56145255/article/details/131731248>

# 设备驱动模型

文件系统---设备数据结构---（总线、设备、驱动）

## 为什么会有驱动模型？

或者说驱动框架

为了统一驱动框架编写规范，让编写驱动更加容易。

设备驱动模型提供了硬件的抽象，内核使用该抽象可以完成很多硬件重复的工作。这样很多重复的代码就不需要编写和调试了，编写驱动程序的难度就有所下降。这些抽象包括如下几个方面：

1. 电源管理
2. 即插即用设备支持：比如热插拔，随时安装和卸载驱动程序
3. 与用户空间的通信：采用虚拟文件系统，对设备文件操作就是对硬件操作。

用户空间程序通过sysfs虚拟文件系统访问设备的相关信息。这些信息被组织成层次结构，用sysfs虚拟文件系统来表示。用户通过对sysfs文件系统的操作，就能控制设备，或者从系统中读出设备的当前信息。

## sysfs文件系统

sysfs文件系统是Linux众多文件系统中的一个。在Linux系统中，每个文件系统都有其特殊的用途。例如ext2用于快速读写存储文件；ext3用来记录日志文件。

Linux设备驱动模型由大量的数据结构和算法组成。这些数据结构之间的关系非常的复杂，多数结构之间通过指针相互关联，构成树形或者网状关系。显示这种关系的最好方法是利用一种树形的文件系统，但是这种文件系统需要具有其他文件系统没有的功能，例如显示内核中的一些关于设备、驱动和总线的信息。为了达到这个目的，Linux内核开发者创建了一种新的文件系统，这就是sysfs文件系统。

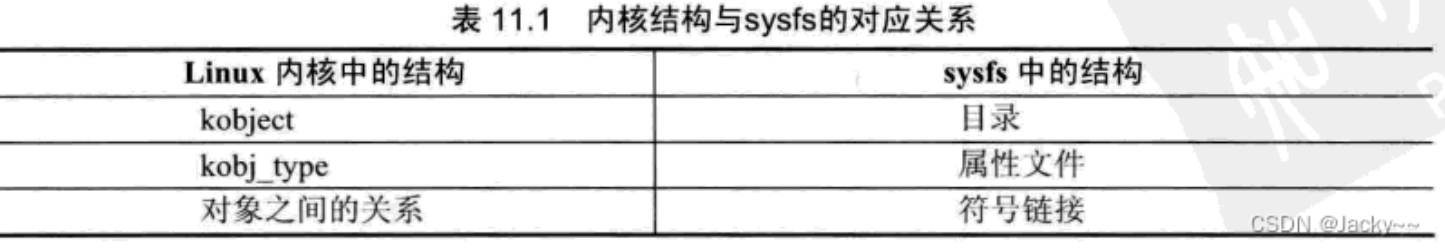
### 1.sys概述

sysfs文件系统是Linux2.6内核的一个新特性，其是一个只存在于内存中的文件系统。内核通过这个文件系统将信息导出到用户空间中。sysfs文件系统的目录之间的关系非常复杂，各目录与文件之间既有树形结构，又有目录关系。

在内核中，这种关系由设备驱动模型来表示。在sysfs文件系统中产生的文件大多数是ASCII文件，通常每个文件有一个值，也可叫属性文件。文件的ASCII码特性保证了被导出信息的准确性，而且易于访问，这些特点使sysfs成为2.6内核最直观，最有用的特性之一。

### 2.sysfs文件系统与内核结构的关系

sysfs文件系统是内核对象（kobject）、属性（kobj\_type）及它们的相互关系的一种表现机制。用户可以从sysfs文件系统中读出内核的数据，也可以将用户空间的数据写入内核中。这是sysfs文件系统非常重要的特性，***通过这个特性，用户空间的数据就能够传送到内核空间中，从而设置驱动程序的属性和状态。***下表揭示了内核中的数据结构与sysfs文件系统的关系。

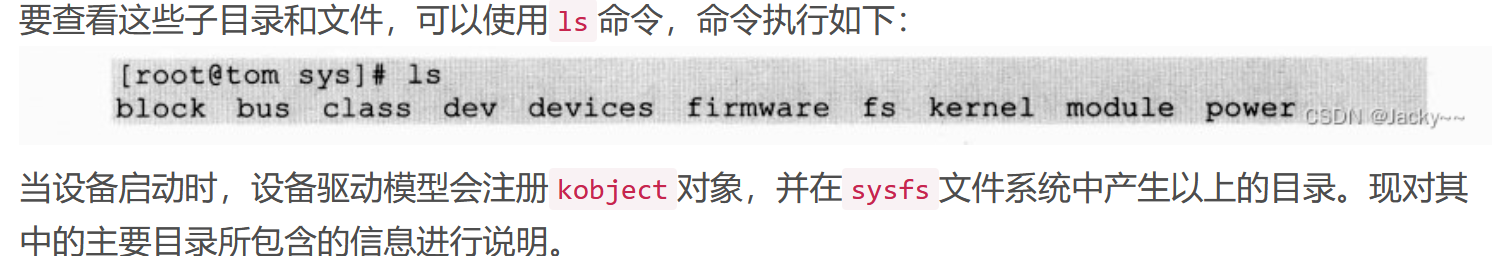


### 3.sysfs文件系统的目录结构

1.sysfs文件系统的目录  
sysfs文件系统与其他文件系统一样，由目录、文件、链接组成。与其他文件系统不同的是，sysfs文件系统表示的内容与其他文件系统中的内容不同。另外，sysfs文件系统只存在于内存中，动态的表示着内核的数据结构。

PSME：内容不同（表示内核，其他文件系统表示硬盘）；存储位置不同（sysfs文件系统只存在于内存）；

sysfs文件系统挂接了一些子目录，这些目录代表了注册sysfs中的主要子系统。



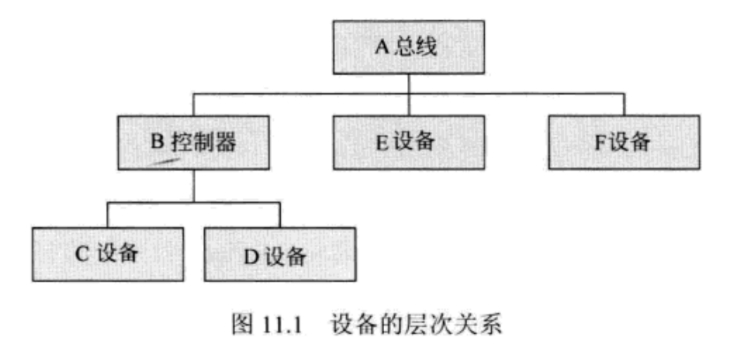
1. block目录：块设备
2. bus目录：物理总线，如：ide\pci\scsi\i2c\pnp
3. class目录

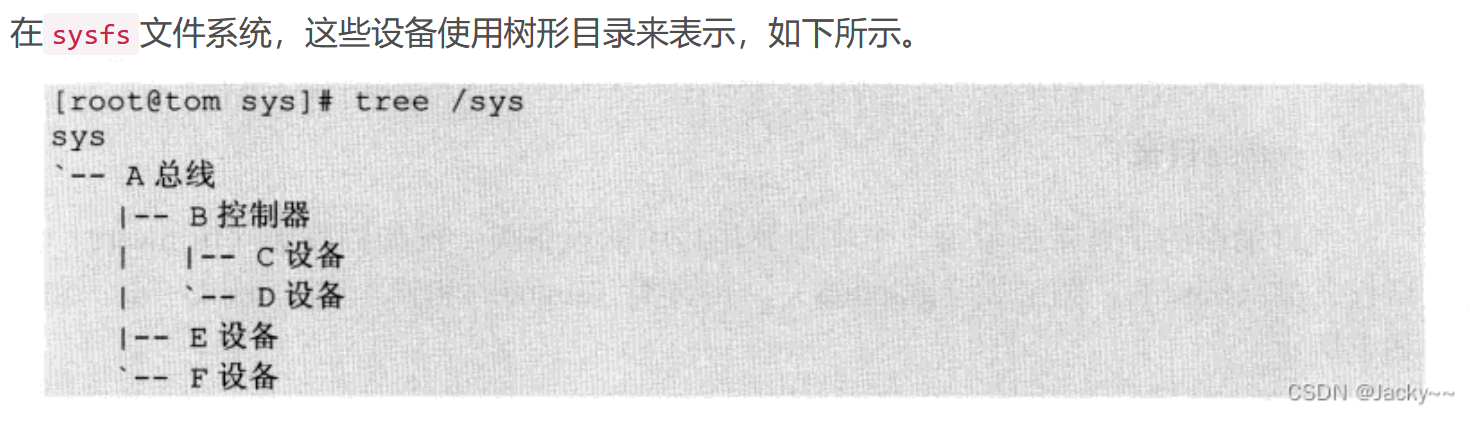
## 设备驱动模型的核心数据结构

设备驱动模型由几个核心的数据结构组成，分别是kobject、kset和subsystem。这些结构使设备驱动模型组成了一个层次结构。该层次结构将驱动、设备和总线等联系起来，形成一个完整的设备模型。下面分别对这些结构进行详细的介绍。

### kobject结构体

宏观上来说，设备驱动模型是一个设备和驱动组成的层次结构。例如一条总线上挂接了很多设备，总线在Linux中也是一种设备，为了表述清楚，这里将其命名为A。在A总线上挂接了一个USB控制器硬件B，在B上挂接了设备C和D，当然如果C和D是一种可以挂接其他设备的父设备，那么在C和D设备下也可以挂接其他设备，但这里认为它们是普通设备。另外在A总线上还挂接了E和F设备，则这些设备的关系如下图所示。



树形结构中每个目录与一个kobject对象相对应，其包含了目录的组织结构和名字等信息。在Linux系统中，kobject结构体是组成设备驱动模型的基本结构。最初它作为设备的一个引用计数使用，随着系统功能的增加，它的任务也越来越多。kobject提供了最基本的设备对象管理能力，每一个在内核中注册的kobject对象都对应于sysfs文件系统中的一个目录。kobject结构体的定义如下：

#### kobject结构体



下面对kobject的几个重要成员介绍如下：

2行是kobject结构体的名称，该名称将显示在sysfs文件系统中，作为一个目录的名字。

6行代表的kobject的属性，可以将属性看成sysfs中的一个属性文件。每个对象都有属性，例如，电源管理需要一个属性表示是否支持挂起；热插拔事件管理需要一个属性来实现设备的状态。因为大部分的同类设备都有相同的属性，因此将这个属性单独组织为一个数据结构kobject\_type，存放在ktype中。这样就可以灵活地管理属性了。需要注意的是，对于sysfs中的普通文件读写操作都是都是由kobject->ktype->sysfs\_ops指针来完成的。对于kobj\_type的详细说明将在后面列出。

第8行的kref字段表示该对象引用的计数，内核通过kref实现对象引用计数管理。内核提供两个函数kobject\_get()、kobject\_put()分别用于增加和减少引用计数，当引用计数为0时，所有该对象使用的资源被释放。下文将对这两个函数详细解释。

第9行的state\_initialized表示kobject是否已经初始化过，1表示初始化，0表示未初始化。unsigend int state\_initialized:1中的1表示，只用unsigned int的最低1位表示这个布尔值。

第10行的state\_in\_sysfs表示kobject是否已经注册到sysfs文件系统中。

2.kobject结构体的初始化函数kobject\_init()

对kobject结构体进行初始化有些复杂。但无论如何，首先应将整个kobject设置为0，一般使用memset()函数来完成。如果没有对kobject置0，那么在以后使用kobject时，可能发生一些奇怪的错误。对kobject置0后，可以调用kobject\_init()函数，对其中的成员进行初始化，该函数的代码如下：

略

# 二、设备驱动模型的三大组件

设备驱动模型有三大重要组件，分别是总线(bus\_type)、设备(device)和驱动(driver)。下面对这三个重要的组件进行分别介绍。

## 总线

从硬件结构上来讲，物理总线有数据总线和地址总线。物理总线是处理器与一个或者多个设备之间的通道。

在设备驱动模型中，所有设备都通过总线连接，此处的总线与物理总线不同，总线是物理总线的一个抽象，同时还包含一些硬件中不存在的虚拟地址总线。在设备驱动模型中，驱动程序是附属在总线上的。下面将首先介绍总线、设备和驱动之间的关系。

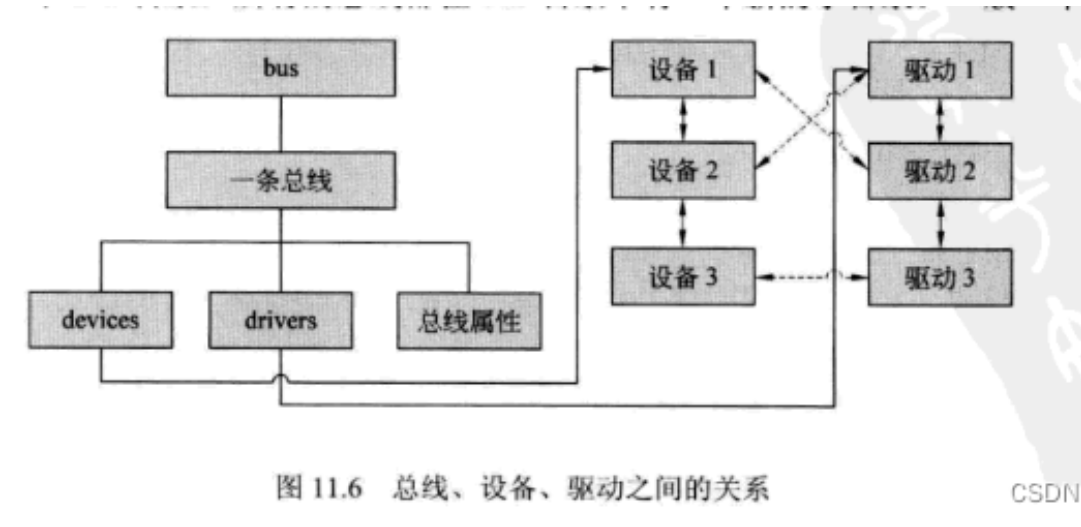
1.总线、设备、驱动关系

在设备驱动模型中，总线、设备和驱动三者之间紧密联系。如下图所示，在/sys目录下，有一个bus目录，所有的总线都在bus目录下有一个新的子目录。一般一个总线目录有一个设备目录、一个驱动目录和一些总线属性文件。

设备目录中包含挂接在该总线上的设备，

驱动目录包含挂接在总线上的驱动程序。

设备和驱动程序之间通过指针互相联系。



如上图所示，总线上的设备链表有3个设备，设备1、设备2和设备3。总线上的驱动链表也有3个驱动程序，驱动1、驱动2和驱动3.其中虚线箭头表示设备与驱动的绑定关系，这个绑定是在总线枚举设备时设置的。这里，设备1与驱动2绑定，设备2与驱动1绑定，设备3与驱动3绑定。

## 设备

在Linux设备驱动模型中，每一个设备都由一个device结构体来描述。device结构体包含了设备所具有的一些通用信息。

对于驱动开发人员来说，当遇到新设备时，需要定义一个新的设备结构体，将device作为新结构体成员。这样就可以在新结构体中定义新设备的一些信息，而设备通用的信息就使用device结构体来表示。使用device结构体的另一个好处是，可以通过device轻松地将新设备加入设备驱动模型的管理中。下面对device结构体进行简要的介绍。

## 驱动

在设备驱动模型中，记录了注册到系统中的所有设备。有些设备可以使用，有些设备不可以使用，原因是设备需要与对应的驱动程序绑定才能使用，本节将重点介绍设备驱动程序。

1.设备驱动device\_driver  
一个设备对应一个最合适的设备驱动程序。但是，一个设备驱动程序就有可能适用多个设备。设备驱动模型自动地探测新设备的产生，并为其分配最合适的设备驱动程序，这样新设备就能够适用了。驱动程序由以下结构体定义：

# 一些概念

## sysfs系统

linux有文件系统，而sysfs系统是虚拟的文件系统，是Linux文件系统的一部分。

设计sysfs系统的目的：方便用户与内核进行通信

虚拟文件系统。提供一种访问内核数据结构的方法，从而使用户空间程序查看和控制系统的设备和资源。

sysfs虚拟文件系统挂载在/sys目录下