驱动绑定

驱动绑定就是一个将能够控制某个设备的设备驱动和该设备联系起来的

过程。因为存在着总线定制结构为代表的设备和驱动，所以总线驱动有

自己的处理方法。而对于那些通用的设备和设备驱动，大部分的绑定操

作都可以使用共同的代码。

总线

~~~

总线类型结构包含了一个在系统中这个总线类型上所有设备的列表。当

为一个设备调用 device\_register 的时候，该设备将被插入到这个列表的

末端。总线对象还包含一个属于该总线类型的所有驱动的列表。当为一

个驱动调用 driver\_register 的时候，该驱动将被插入到这个列表的末端。

这就是能够触发驱动绑定两个事件。

设备注册

~~~~~~~~~~~~~~~

当添加一个新设备时，会迭代总线的驱动列表，以找到一个支持该设备

的驱动。为了确保这点，该设备的 ID 必须和那个驱动所支持的设备 ID

中的一个相匹配。 而比较 ID 的格式和语义则都是总线定制的。它是由

总线驱动提供的一个回调函数来比较设备和驱动，而不是试图推导出一

个复杂的状态机和匹配算法。

int match(struct device \* dev, struct device\_driver \* drv);

如果匹配成功，设备的 driver 字段会设置成指向该驱动，然后就调用该驱

动的 probe 回调函数。这将给这个驱动一个机会去确认它的确支持这个硬

件，并且该硬件正处于工作状态中。

设备的 class

~~~~~~~~~~~~

在成功完成 probe 的时候，设备也与它所属的 class 注册完成。设备驱动属

于一个且唯一一个 class，它被设置在驱动的 devclass 字段。在调用 class 的

回调函数 register\_dev 时，devclass\_add\_device 会被调用来枚举属于该 class

的设备，并真正的用 class 将设备进行注册。

驱动

~~~~~~

当一个驱动连接到一个设备时，该设备就被插入到这个驱动的设备列表里去。

系统文件系统

~~~~~

在总线的‘devices’目录中会创建一个符号链接，该链接指向在物理层次中的设

备目录。

在驱动的‘devices’目录中会创建一个符号链接，该链接指向在物理层次中的设

备目录。

在 class 的目录中会为该设备创建一个目录。然后会在该目录中创建一个符号

链接，该链接指向这个设备在 sysfs 树中的物理位置。

在设备的物理目录下，都可以创建一个符号链接（即使这还没有完成）来指向

它的 class 目录或 class 的顶层目录。同时也可以创建指向它的驱动目录的符号

链接。

驱动注册

~~~~~~~~~~~~~~~

当添加一个新驱动时，也几乎是相同的流程。将迭代总线上的设备列表将用来找

到一个匹配。那些已经绑定的驱动的设备将被忽略。所有的设备都将参加迭代，

目的是为这个新驱动绑定尽量多的设备。

移除

~~~~~~~

当移除一个设备时，它的引用计数也最终会变成 0。这时，驱动的回调函数 remove

会被调用。它将从驱动的设备列表里移除这个设备并递减该驱动的引用计数。最后，

这两者之间的符号链接都将被移除。

当移除一个驱动时，将迭代它所支持的设备列表，并对每一个设备调用驱动的 remove

回调函数。设备将从那个列表中移除，然后符号链接也将被移除。