Interface

设备接口
介绍
设备接口是指直接与用户空间相连的设备类逻辑接口，比如说设备节点。
每一个设备类可能拥有多个的接口，通过这些接口，你可以使用同样的设备。输入设备可能支持鼠标接口、evdev界面和触摸屏界面。SCSI磁盘将支持磁盘接口，一般的SCSI接口，或者可能是原始设备接口。
设备接口被注册到他们所属的类。随着设备被添加到类，它们便被添加到每个接口注册的类别。该接口负责决定是否该设备支持这个接口或者没有。
编程接口
～～～～～～～～～～～～

```
struct device_interface {
        char                * name;
        rwlock_t              lock;
        u32                 devnum;
        struct device_class     * devclass;

        struct list_head        node;
        struct driver_dir_entry dir;

        int (*add_device)(struct device *);
        int (*add_device)(struct intf_data *);
};

int interface_register(struct device_interface *);
void interface_unregister(struct device_interface *);
```

一种接口必须指定他所属的设备类。它被添加到该类别接口的注册清单上。
接口可能随时加入到一种类别中。一旦它被加入，这一类别中的每一设备通过界面的 add_device 回调。当一种接口注销后，每一种设备从接口上移开。

设备
～～～～～～～～

一旦一个设备加入一个设备类中（class），它将同时被注册个这个类的每个接口。每个设备都需要在device::class_data中填写和设备相关的信息。每个接口都能够使用这些信息来判断该设备是否支持该接口。

数据
～～～～～～

```
struct intf_data {
       struct list_head       node;
       struct device_interface * intf;
       struct device          * dev;
       u32                    intf_num;
};

int interface_add_data(struct interface_data *);
```

该接口是负责分配并初始化一种结构intf_data，并调用interface_add_data()将其添加到他所属于的设备接口清单中。当设备从类中注销时该清单会被更新（而不是一个类中所有可能的接口（发生改变））。这种结构或许应该根植于无论何种的单一设备数据结构，无论接口是如何分配的。
枚举设备的接口。其发生在interface_add_data()和枚举值中，存放在设备的结构intf_data。

sysfs
～～～～～

在其所属的设备类的目录中，每一种接口都给定一索引：
在class目录中，接口同时得到一个索引：

```
    class/
    `-- input
        |-- devices
```

```
        |-- drivers
        |-- mouse
        `-- evdev
```
当一个设备加入到接口中，便形成了一个指向物理层中设备目录的符号链接。
```
class/
   `-- input
       |-- devices
       |   `-- 1 -> ../../../root/pci0/00:1f.0/usb_bus/00:1f.2-1:0/
       |-- drivers
       |   `-- usb:usb_mouse -> ../../../bus/drivers/usb_mouse/
       |-- mouse
       |   `-- 1 -> ../../../root/pci0/00:1f.0/usb_bus/00:1f.2-1:0/
       `-- evdev
           `-- 1 -> ../../../root/pci0/00:1f.0/usb_bus/00:1f.2-1:0/
```

未来计划
～～～～～～
一种设备接口直接与一个用户空间的接口相连，（一种特殊的设备节点。）比如说，一个SCSI磁盘提供了至少两个用户空间的接口：标准的SCSI磁盘接口和一般的SCSI接口。他可能也会使用一个更加原始设备接口。

许多接口都具有一个主设备号，该接口下的每个设备都有一个次设备号。 在特殊情况下，几个接口共用一个主设备号，分别使用不同区间的从设备号（比如说是输入设备）。

我们可以将主设备号和次设备号保存在interface结构体中。在接口向class注册时，将分配主设备号和次设备号。我们也可以使用一个内核函数来完成主次设备号的分配。