

“菜鸟玩转嵌入式”视频培训讲座

— Linux驱动开发基础班

主办：上海申嵌信息科技有限公司

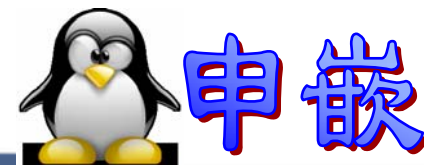
承办：嵌入式家园

协办：上海嵌入式家园-开发板商城
广州友善之臂计算机科技有限公司

主讲：贺光辉（嵌入式系统工程师）

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



Linux驱动开发基础

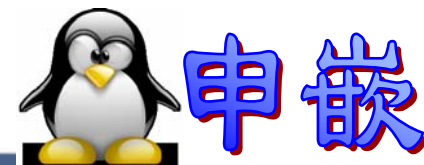
嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

- 掌握嵌入式Linux设备驱动程序的基本原理、架构和设计方法
 - 字符设备驱动
 - 块设备驱动
 - 网络设备驱动
- 掌握Linux设备驱动开发中常用的机制和内核资源
 - 中断顶/底半部处理
 - 内核定时器和延时操作
 - 并发控制在内核中的应用
 - 内存管理和分配
 - 阻塞型I/O和非阻塞型I/O

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



第一章

Linux驱动程序概述

主讲：贺光辉

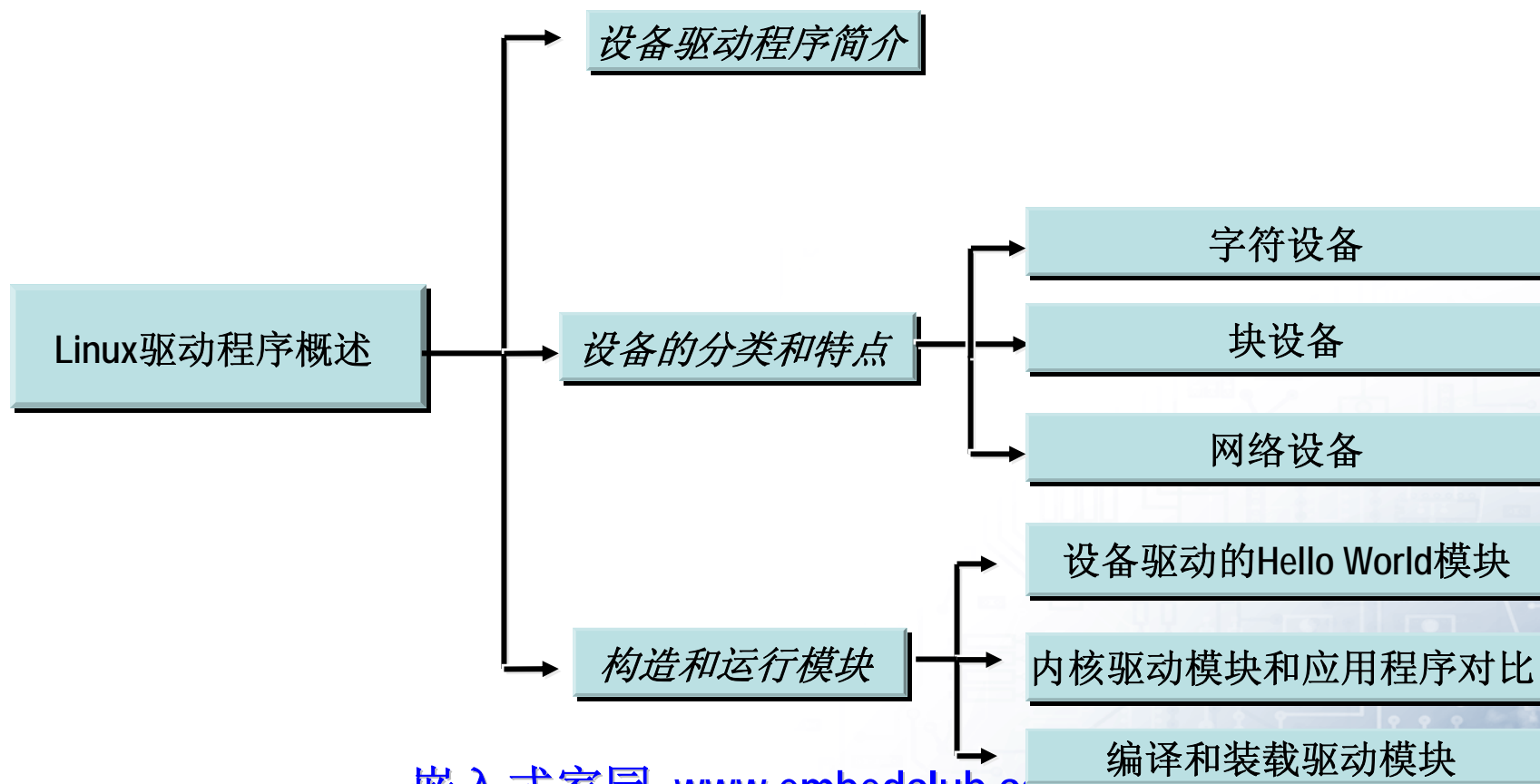
嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

- 了解Linux设备驱动程序开发的基础知识
- 掌握Linux驱动模块的构造和装载方法

嵌入式家园 www.embedclub.com

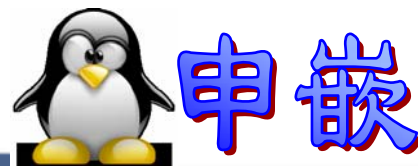
上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

为什么要学习嵌入式Linux驱动程序开发？



● 高需求

- 内核代码的大部分
- 新芯片、新设备

● 高门槛

- 需要具有硬件知识
- 需要了解内核基础知识
- 需要了解内核中的并发控制和同步
- 复杂的软件结构框架

● 高回报

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

- 操控硬件，是应用程序和硬件设备之间的一个接口
 - 隐藏硬件细节，提高应用软件的可移植性
- 提供机制，而不是提供策略
 - 机制：驱动程序能实现什么功能
 - 策略：用户如何使用这些功能

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

● 设备分类

- 字符设备(char device)
- 块设备(block device)
- 网络设备(network device)

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

● 字符设备特点

- 像字节流一样来存取的设备(如同文件), 对它的读写是以字节为单位。
 - 比如: 串口在进行收发数据时就是一个字节一个字节进行
- 通常至少需要实现 open, close, read和 write 等系统调用
- 应用程序可以通过/dev下的文件系统结点来访问字符设备。
 - 比如: /dev/led, /dev/ttySAC0
- 只能顺序访问的数据通道, 不能前后移动访问指针。
 - 特例: 比如framebuffer设备就是这样的设备, 应用程序可以使用mmap或lseek访问图像的各个区域

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

● 块设备特点

- 块设备上的数据以块的形式存放，
 - 比如Nand Flash上的数据就是以页为单位存放的。
- 块设备驱动程序向用户层提供的接口与字符设备一样，应用程序也可以通过对应的设备文件来调用open/close/read/write等系统调用，与块设备传送任意字节的数据。
 - 比如/dev/mtdblock0
- 块设备和字符设备的区别仅仅在于内核内部管理数据的方式
 - 以块为单位实现数据的读/写
 - 块设备按照一定的格式存放数据，不同的文件系统类型就是用来定义这些格式的。
- 块设备驱动程序必须向内核的文件系统提供专门的接口，而这些接口使得在块设备上存放文件系统，挂载（mount）块设备。

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

● 网络设备特点

- 特殊的一类设备，传输的数据是有结构的、成块的（报文、包、帧），但是块的大小又不是固定的，大到数百甚至数千字节，小到几个字节。
- 任何一个网络事务都通过一个网络接口，即一个能够和其他主机交换数据的设备。
 - 网卡
 - 软件设备：回环接口（loopback）
- 一个网络接口负责发送和接收数据报文
- 内核调用一套和数据包传输相关的函数与网络设备交互

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

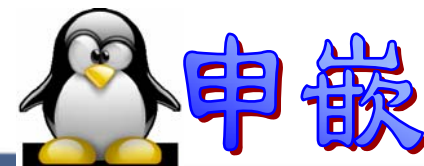
● 驱动程序加入内核的方法

- 把所有需要的功能都加入到内核中
 - 生成的内核镜像（Image）文件
 - 如果我们要在现有的内核中增加或删除功能，将不得不重新编译和装载内核。

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3驱动程序加入内核的方法



- Linux提供被称为**模块（Module）**的机制
 - 提供了对许多模块支持, 包括但不限于设备驱动
 - 每个模块由目标代码组成(没有连接成一个完整可执行程序)
 - insmod 将模块动态加载到**正在运行的内核**
 - rmmod 程序移除模块

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3-1 设备驱动的Hello World模块(hello.c)



申嵌

```
#include <linux/init.h>
#include <linux/module.h>
```

自由许可证

```
MODULE_LICENSE("Dual BSD/GPL");
```

```
static int __init hello_init(void)
```

```
{
    printk(KERN_ALERT "Hello world\n");
    return 0;
}
```

用法类似于printf，但它有优先级(比如 KERN_ALERT)

```
static void __exit hello_exit(void)
```

```
{
    printk(KERN_ALERT " Hello world exit\n");
}
```

宏，告诉内核这两个函数只会在加载和卸载模块时使用

```
module_init(hello_init);
```

```
module_exit(hello_exit);
```

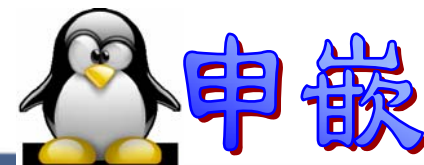
模块初始化宏

嵌入式家园 www.embedclub.com

模块卸载宏

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3-1 设备驱动的Hello World模块(hello.c)



- 编译内核模块

`$(MAKE) -C $(KERNELDIR) M=$(PWD) modules`

- 加载内核模块:

- `insmod ./hello.ko`

- 查看内核中已装载的模块

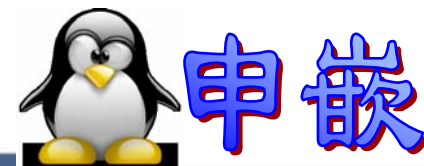
- `lsmod | grep hello`

- 卸载内核模块:

- `rmmmod hello`

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



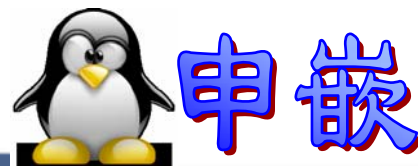
Linux内核模块的程序结构

- **module_init()---模块加载函数（必须）**
 - 通过insmod或modprobe命令加载内核模块时，模块的加载函数会自动被内核执行，完成模块的相关初始化工作
- **module_exit()---模块卸载函数（必须）**
 - 当通过rmmod命令卸载某模块时，模块的卸载函数会自动被内核执行，完成与模块装载函数相反的功能

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3-1 设备驱动的Hello World模块(hello.c)

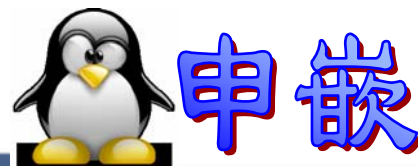


- **MODULE_LICENSE()---模块许可证声明（必须）**
 - 模块许可证（LICENSE）声明描述内核模块的许可权限
 - 如果不声明LICENSE,模块被加载时，将收到内核被污染（kernel tainted）的警告
- **module_param()---模块参数（可选）**
 - 模块参数是模块被加载的时候可以被传递给它的值，它本身对应模块内部的全局变量。

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3-1 设备驱动的Hello World模块(hello.c)

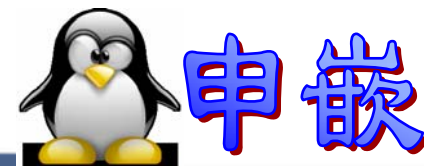


- **EXPORT_SYMBOL()---模块导出符号（可选）**
 - 内核模块可以导出符号（symbol，对应于函数或变量）到内核
 - 其他模块可以使用本模块中的变量或函数
- 其他一些声明**MODULE_XXXXX()---模块声明（可选）**

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3-1 设备驱动的Hello World模块



● 模块加载函数

```
static int __init initialization_function(void)
{
    /* 初始化代码 */
}
module_init(initialization_function);
```

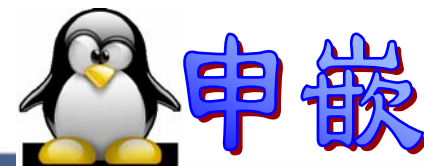
应当声明成静态的 (static), 因为它们不会在特定文件之外可见

表明该函数只是在初始化时使用。模块加载后会丢掉这个初始化函数, 这样可以释放出来, 以作他用。

定义会在模块目标代码中增加一个特殊的段, 用于说明内核模块初始化函数所在的位置。没有这个定义, 初始化函数不会被调用。

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



● 模块卸载函数

```
static void __exit cleanup_function(void)
{
    /* 释放资源 */
}
module_exit(cleanup_function);
```

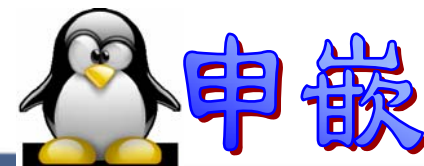
在模块被移除前注销接口并释放所有所占用的系统资源

标识这个代码是只用于模块卸载或系统停止时调用，其他任何时候调用都是错误。

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3-1 设备驱动的Hello World模块

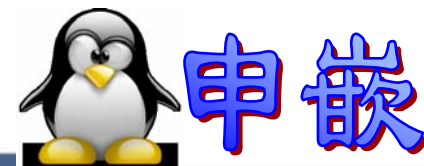


```
#include <linux/init.h>
#include <linux/module.h>
MODULE_LICENSE("Dual BSD/GPL");
static char *whom = "world";
static int howmany = 1;
static int hello_init(void)
{
    int i;
    for(i=0;i<howmany;i++){
        printk(KERN_ALERT "Hello %s\n",whom);
    }
    return 0;
}
static void hello_exit(void)
{
    printk(KERN_ALERT " Hello world exit\n");
}
module_init(hello_init);
module_exit(hello_exit);
module_param(howmany, int, S_IRUGO);
module_param(whom, charp, S_IRUGO);
```

module_param (参数名,
参数类型, 参数读/写权
限)

嵌入式家园 www.embedclub.com

上海嵌入式家园 开发板商城 <http://embedclub.taobao.com/>



module_param（参数名，参数类型，参数读/写权限）

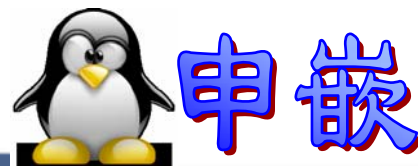
```
static char *whom = "world";  
static int howmany = 1;  
module_param(howmany, int, S_IRUGO);  
module_param(whom, charp, S_IRUGO);
```

内核支持的模块参数类型包括：

byte、short、ushort、int、uint、long、ulong、
charp(字符指针)、bool,以‘u’开头的为无符号值。

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



● 模块也可以拥有参数数组

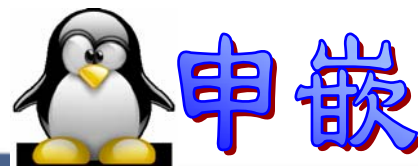
- 形式为“module_param_array（数组名，数组类型，数组长，参数读/写权限）”。
- 运行insmod或modprobe命令时，应使用逗号分隔输入的数组元素

● 装载模块时改变参数：

- 可通过insmod或modprobe
- insmod hello_ext.ko howmany=5 whom="Students"
- modprobe也可以从它的配置文件(/etc/modprobe.conf)读取参数的值

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



● 模块导出符号

- `EXPORT_SYMBOL(name);`
- `EXPORT_SYMBOL_GPL(name);`

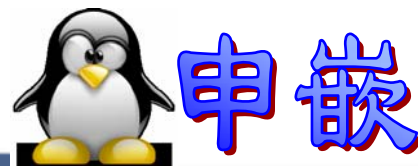
_GPL 版本的宏定义的导出符号只能对 GPL 许可的模块可用

● 内核符号表: `/proc/kallsyms` 记录内核模块中导出符号及所在的内核地址

符号必须在模块文件的全局部分导出, 不能在函数中导出

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



● 模块声明与描述

MODULE_AUTHOR(author); ---声明模块的作者

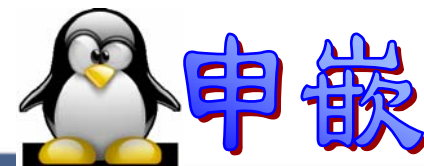
MODULE_DESCRIPTION(description); ---声明模块的描述

MODULE_VERSION(version_string); ---声明模块的版本

MODULE_ALIAS(alternate_name); ---声明模块的别名

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



● 模块的使用计数

● Linux2.4内核

- MOD_INC_USE_COUNT (加一计数)
- MOD_DEC_USE_COUNT (减一计数)

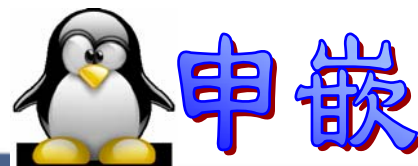
● Linux2.6内核中

- `int try_module_get(struct module *module);`
- `void module_put(struct module *module);`

在Linux2.6内核下，对于设备驱动工程师而言，很少需要亲自调用try_module_get()和module_put()，因为模块的计数管理由内核里更底层的代码（如总线驱动或是此类设备共用的核心模块）来实现，从而简化了设备驱动的开发

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



● 应用程序是一个进程

- 编程从主函数main()开始
- 主函数main返回即是进程结束

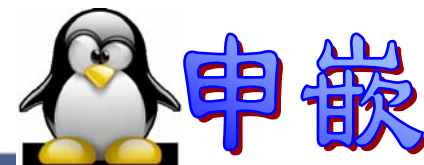
● 驱动程序是一系列内核函数

- 驱动程序向内核添加了一些函数,是内核的一部分
 - open ()
 - release ()
 - read ()
 - write ()
- 这些函数由内核在适当的时候来调用
- 这些函数可以用来完成硬件访问等操作

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

1-3-3 编写模块Makefile文件



Makefile文件

```
ifneq ($(KERNELRELEASE),)
obj-m := hello.o
else
KERNELDIR ?= /home/student/linux-2.6.32.2 PWD := $(shell pwd)
default:
$(MAKE) -C $(KERNELDIR) M=$(PWD) modules
endif
```

如果我们想由两个源文件(比如file1.c和file2.c)构造出一个名称为module.ko的模块, 则正确的makefile可如下编写:

```
obj-m := module.o
module-objs := file1.o file2.o
```

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

● 装载模块

- insmod和modprobe可以用来装载模块

insmod和modprobe主要区别

modprobe会考虑要装载的模块是否引用了一些当前内核不存在的符号。如果有这类引用，modprobe会在当前模块路径中搜索定义了这些符号的其他模块，并同时将这些模块也装载到内核。如果在这种情况下使用insmod，该命令则会失败，并在系统日志文件中记录“unresolved symbols（未解析的符号）”消息。

● 卸载模块

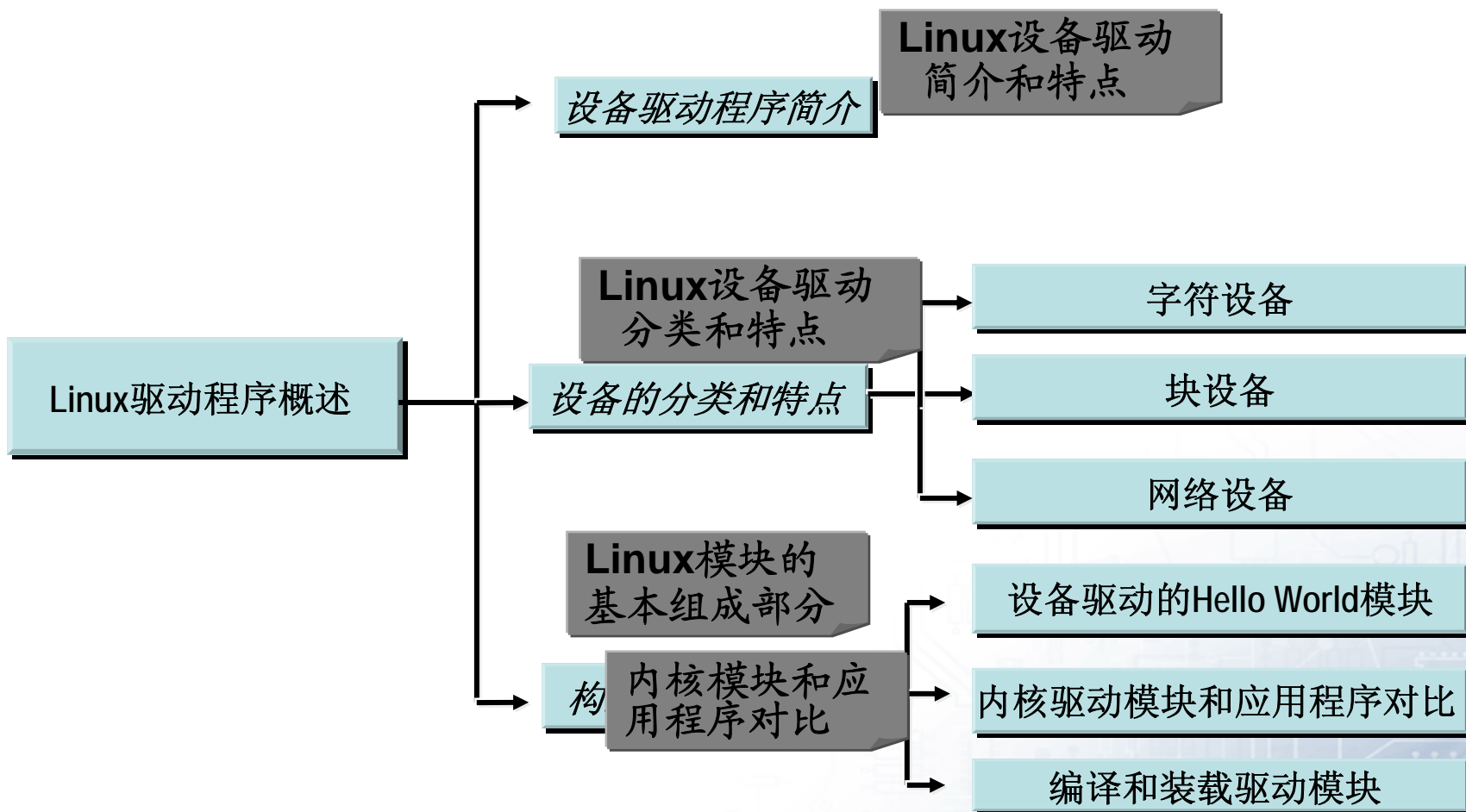
- 从内核中卸载模块可以用rmmod工具

注意，如果内核认为该模块仍然在使用状态，或者内核被禁止移除该模块，则无法移除该模块。

- Linux驱动模块的基本结构
- Linux驱动模块的编译方法
- Linux驱动模块的装载和卸载方法

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>



嵌入式家园 www.embedclub.com

如何编译内核模块
如何装载和卸载内核模块

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>

- 任务1: **Hello World** 模块程序设计
- 任务2: 带传入参数的**Hello World** 模块程序设计

嵌入式家园 www.embedclub.com

上海嵌入式家园-开发板商城 <http://embedclub.taobao.com/>