

内核操作Linux2.6内核驱动移植参考 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/206/2021\\_2022\\_\\_E5\\_86\\_85\\_E6\\_A0\\_B8\\_E6\\_93\\_8D\\_E4\\_c103\\_206586.htm](https://www.100test.com/kao_ti2020/206/2021_2022__E5_86_85_E6_A0_B8_E6_93_8D_E4_c103_206586.htm) 随着Linux2.6的发布，由于2.6内核做了教的改动，各个设备的驱动程序在不同程度上要进行改写。为了方便各位Linux爱好者我把自己整理的这份文档share出来。该文当列举了2.6内核同以前版本的绝大多数变化，可惜的是由于时间和精力有限没有详细列出各个函数的用法。 特别声明：该文档中的内容来自<http://lwn.net>，该网也上也有各个函数的较为详细的说明可供各位参考。 1

、使用新的入口 必须包含 `< linux/init.h >`

`module_init(your_init_func). module_exit(your_exit_func).` 老版本：`int init_module(void). void cleanup_module(voi).` 2.4中两种都可以用，对如后面的入口函数不必要显示包含任何头文件。

2、`GPL MODULE_LICENSE("Dual BSD/GPL").` 老版本：`MODULE_LICENSE("GPL").` 3、模块参数 必须显式包含

`< linux/moduleparam.h > module_param(name, type, perm).`  
`module_param_named(name, value, type, perm).` 参数定义  
`module_param_string(name, string, len, perm).`

`module_param_array(name, type, num, perm).` 老版本：`MODULE_PARM(variable,type).`

`MODULE_PARM_DESC(variable,type).` 4、模块别名

`MODULE_ALIAS("alias-name").` 这是新增的，在老版本中需在`/etc/modules.conf`配置，现在在代码中就可以实现。 5、模块计数

`int try_module_get(&module). module_put().` 老版本：`MOD_INC_USE_COUNT` 和 `MOD_DEC_USE_COUNT` 6、

符号导出 只有显示的导出符号才能被其他模块使用，默认不导出所有的符号，不必使用EXPORT\_NO\_SYMBOLS 老板本：默认导出所有的符号，除非使用EXPORT\_NO\_SYMBOLS 7、 内核版本检查 需要在多个文件中包含 < linux/module.h > 时，不必定义\_\_NO\_VERSION\_\_ 老版本：在多个文件中包含 < linux/module.h > 时，除在主文件外的其他文件中必须定义\_\_NO\_VERSION\_\_，防止版本重复定义。 8、 设备号 kdev\_t被废除不可用，新的dev\_t拓展到了32位，12位主设备号，20位次设备号。 unsigned int iminor(struct inode \*inode). unsigned int imajor(struct inode \*inode). 老版本：8位主设备号，8位次设备号 int MAJOR(kdev\_t dev). int MINOR(kdev\_t dev). 9、 内存分配头文件变更 所有的内存分配函数包含在头文件 < linux/slab.h >，而原来的 < linux/malloc.h > 不存在 老版本：内存分配函数包含在头文件 < linux/malloc.h > 10、 结构体的初试化 gcc开始采用ANSI C的struct结构体的初始化形式：static struct some\_structure = { .field1 = value, .field2 = value, .. }. 老版本：非标准的初试化形式 static struct some\_structure = { field1: value, field2: value, .. }. 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)