

1 编写 Hello_World 模块

首先编写 hello.c，代码如下：

```
1  #include <linux/init.h>
2  #include <linux/module.h>
3  MODULE_LICENSE("Dual BSD/GPL");
4
5  static int hello_init(void)
6  {
7      printk(KERN_ALERT "Hello, world\n");
8      return 0;
9  }
10
11 static void hello_exit(void)
12 {
13     printk(KERN_ALERT "Goodbye, cruel world\n");
14 }
15
16 module_init(hello_init);
17 module_exit(hello_exit);
```

写一个简单的 Makefile 文件：

```
1  obj-m := hello.o
2  KERNELDIR := /lib/modules/$(shell uname -r)/build
3  PWD := $(shell pwd)
4
5  modules:
6      $(MAKE) -C $(KERNELDIR) M=$(PWD) modules
```

2 Hello_World 模块的代码解析

2.1 解析 hello.c

hello.c 中的代码很简单，这里主要说一下用到的一些 API 函数。

module_init() 是一个宏，用于指定模块初始化的函数，所以 hello_init() 函数在模块装载到内核时会被调用。module_exit() 是一个宏，用于指定模块的清除函数，所以 hello_exit() 函数从内核卸载时会被调用。

MODULE_LICENSE() 是一个宏，用来告诉内核该模块使用的许可证。如果没有使用 MODULE_LICENSE("Dual BSD/GPL")，内核在装载该模块时会产生抱怨。

printk() 函数和 printf() 函数功能类似，但 printk() 函数是内核函数。当模块连接到内核时，就可以调用 printk() 函数。

2.2 解析 Makefile

首先说明，这里不会介绍 Makefile 语法，而是介绍为什么 Makefile 的内容是这些。

“obj-m := hello.o”，用于说明一个模块需要从目标文件 hello.o 中构造，而从该目标文件构造的模块名称为 hello.ko。

“KERNELDIR := /lib/modules/\$(shell uname -r)/build” 是给变量赋值，KERNELDIR 的值为当前内核源代码目录。“PWD := \$(shell pwd)” 中 PWD 的值为当前目录。

“\$(MAKE) -C \$(KERNELDIR) M=\$(PWD) modules” 这个命令是真正用来构造模块的。命令首先改变目录到 -C 选项指定的位置，在这个目录中有内核自己的 Makefile 文件，这里的 Makefile 文件用于真正构造模块。等到该 Makefile 要真正地构造 modules 目标时，返回到 -M 选项指定的目录，modules 目标指向的是 obj-m 变量中设定的模块。

也就是说，当我们执行“make modules”的时候，将执行“\$(MAKE) -C \$(KERNELDIR) M=\$(PWD) modules”。make 命令将首先进入 -C 选项指定的内核源码树，运行内核构造系统，其中 Makefile 要真正构造模块之前，make 命令又回到 -M 选项指定的目录，在该目录下生成 obj-m 指定的模块。

3 使用 Hello_World 模块

```
1 # 编译模块
2 pengsida@scholes:~/ldd$ make modules
3 make -C /lib/modules/4.8.0-22-generic/build M=/home/pengsida/ldd modules
4 make[1]: Entering directory '/usr/src/linux-headers-4.8.0-22-generic'
5 Building modules, stage 2.
6 MODPOST 1 modules
7 make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-22-generic'
8
9 # 安装 hello 模块
10 pengsida@scholes:~/ldd$ sudo insmod ./hello.ko
11
12 # 查看 hello 模块是否在代码中
13 pengsida@scholes:~/ldd$ lsmod | grep hello
14 hello                16384  0
15
16 # 卸载 hello 模块
17 pengsida@scholes:~/ldd$ sudo rmmod hello
18
19 # 查看 hello 模块运行时输出的信息
20 pengsida@scholes:~/ldd$ cat /var/log/syslog | grep world
21 Apr 23 21:32:49 scholes kernel: [ 1779.811190] hello world
22 Apr 23 21:33:40 scholes kernel: [ 1830.907433] Goodbye, cruel world
```