有时候写了一些内核模块，功能简单并且和内核的其它模块联系不多的时候（这个联系主要是指内核其它模块不要调用这个模块中的一些函数），可以把模块写成可加载的形式，但是有时候写的内核模块是用来被其它模块调用的时候，就不能写成可加载的形式了，要把这个模块编译到内核里面去

拿NETLINK来说，在dpi设备demo2的linux系统中，驱动把包传给协议栈后，协议栈要把包传到用户空间以供分析，如果采用netlink socket的方法来传数据包到用户空间的话，协议栈里就要调用自己写的netlink的函数send2user，该函数负责把数据包通过netlink socket传到用户空间，该模块的具体代码见kernelspcae.c

下面开始将怎么把kernelspace.c这个模块编译到内核里面去

1. 在内核源代码的drivers目录下创建文件夹gupeng



1. 把kernelspace.c移入gupeng文件夹中，并创建新文件Kconfig,Makefile



1. 编辑Kconfig,内容如下

menu "gupeng's modules"

config NETLINK\_KMODULE

tristate "netlink kernelspace module"

endmenu

(4) Makefile文件编辑如下

obj-$( CONFIG\_NETLINK\_MODULE) +=kernelspace.o

注意红字的区别，Makefile中的要比Kconfig中的多一个CONFIG\_

(5) 修改父文件夹中的Kcofig,即drivers目录下的Kconfig,添加如下内容：

source "drivers/gupeng/Kconfig"

表示引用gupeng/Kconfig这个新的Kconfig文件

（6）修改父文件夹中的Makefile,添加如下内容

obj-$( CONFIG\_NETLINK\_MODULE) +=gupeng/

使得用户在编译的时候进入gupeng目录，执行里面的Makefile

(7)开始配置，执行make menuconfig,进入device drivers目录

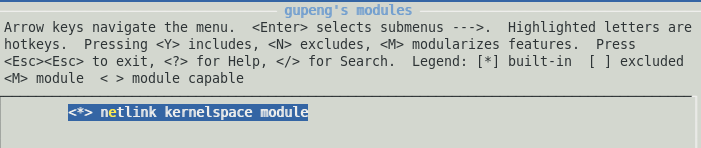


在该目录下可以找到刚刚添加到配置文件中的模块的选项



（8）进入该选项，看到可以配置的信息，这是Kconfig中tristate netlink kernel module

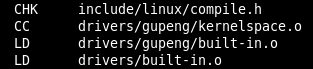
的结果



（9）选中改选项，将它编译进内核，退出menuconfig

(10)编译内核：make

可以看到程序被编译进内核的信息



就这样，我们自己写的一个程序就被编译进内核

编译出来的结果：

