**Linux模块makefile写法**

obj-m+=hello.o

all:

make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules

clean:

make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) clean

这个makefile的作用就是编译hello.c文件，最终生成hello.ko文件。

1.

obj-m表示编译生成可加载模块。

obj-m+=hello.o，这条语句就是显式地将hello.o编译成hello.ko,而hello.o则由make的自动推导功能编译hello.c文件生成。

2.

all，clean这一类的是makefile中的伪目标，伪目标并不是一个真正的编译目标，它代表着一系列你想要执行的命令集合

3.

标准的make指令是这样的：make -C $KDIR M=$PWD [target]

①:-C选项：此选项指定内核源码的位置，make在编译时将会进入内核源码目录，执行编译，编译完成时返回。

②:$KDIR：/lib/modules/$(shell uname -r)/build/，指定内核源码的位置。

直接在目标板上编译时，内核头文件默认存放在/lib/modules/$(shell uname -r)/build/中，这个build/目录是一个软连接，链接到源码头文件的安装位置。而内核真正的源码库则直接引用正在运行的内核镜像。

③:M=$(PWD)或M=$(CURDIR)：需要编译的模块源文件地址,即hello.c文件的地址

④:[target]：modules，事实上，这是个可选选项。默认行为是将源文件编译并生成内核模块