**Linux内核makefile**

1. **make和makefile**

make是linux的一个程序，针对makefile文件，执行make，他会自动去找**当前目录**下的makefile文件。

linux内核源代码的编译也是使用make工具和makefile，但是它在普通的C程序编译的基础上对配置和编译选项进行了扩展，这就是kbuild系统，专门针对linux的内核编译，使得linux内核的编译更加简洁而高效。

1. **多种makefile**

由于linux内核分为多层，并且兼容多平台的特性，makefile就可以在各个目录中，对各层模块进行灵活的分离编译，就可以减低耦合性。

makefile五个部分：



* 将模块编译进内核

obj-y += foo.o

将foo.o编译进内核中，make自动将foo.c编译为foo.o，这个方式为模块单独编译

需要一次编译整个内核时可以这样写：

obj-$(CONFIG\_FOO) += foo.o

不过，要在.config文件中将CONFIG\_FOO变量配置成y

* 将模块编译为可加载模块

obj-m += foo.o

同样的，也可这样写：obj-$(CONFIG\_FOO) += foo.o，也需要在.config文件中将CONFIG\_FOO变量配置成m

**Makefile语法规则**

**赋值规则：**

1. **“=”**

**变量的值为整个Makefile文件中最后被指定的值。**

1. **“:=”**

**直接赋值**

1. **“?=”**

**如果该变量没有被赋值，则赋予等号后的值。**

1. **“+=”**

**表示将等号后面的值添加到前面的变量上。**