



## C语言高级编程-Makefile二

主讲：小美老师

创客引领未来

扫微信二维码 获取更多信息





Lemon

昵称：小姜老师  
华清创客学院，嵌入式讲师

makerU

- 直接运行make
- 选项
  - -C dir读入指定目录下的Makefile
  - -f file读入当前目录下的file文件作为Makefile
  - -i忽略所有的命令执行错误
  - -I dir指定被包含的Makefile所在目录
  - ✓ – -n只打印要执行的命令，但不执行这些命令
  - ✓ – -p显示make变量数据库和隐含规则
  - ✓ – -s在执行命令时不显示命令
  - -w如果make在执行过程中改变目录，打印当前目录名

# Makefile的隐含规则

- 隐含规则1：编译C程序的隐含规则
- “<n>.o” 的目标的依赖目标会自动推导为 “<n>.c”，并且其生成命令是 “\$(CC) -c \$(CPPFLAGS) \$(CFLAGS)”



# Makefile的隐含规则

- 隐含规则2：链接Object文件的隐含规则
- “<n>” 目标依赖于 “<n>.o”，通过运行C的编译器来运行链接程序生成（一般是“ld”），其生成命令是：“\$(CC) \$(LDFLAGS) <n>.o \$(LOADLIBES) \$(LDLIBS)”。这个规则对于只有一个源文件的工程有效，同时也对多个Object文件（由不同的源文件生成）的也有效。例如如下规则：
  - `x : x.o y.o z.o`
  - 并且“x.c”、“y.c”和“z.c”都存在时，隐含规则将执行如下命令：
    - `cc -c x.c -o x.o`
    - `cc -c y.c -o y.o`
    - `cc -c z.c -o z.o`
    - `cc x.o y.o z.o -o x`
  - 如果没有一个源文件（如上例中的x.c）和你的目标名字（如上例中的x）相关联，那么，你最好写出自己的生成规则，不然，隐含规则会报错的。



扫微信二维码 获取更多信息