Make的使用

# 1 make介绍­

## 1.2 make组成

Make由5部分组成，显示规则、隐含规则、变量定义、makefile指示符和注释。

显示规则 – 主动编写描述规则，即明确指出各目标文件的所依赖的的源文件集合，以及编译目标文件所需要的命令；

隐含规则 – make中默认规则，自动推导目标文件的依赖关系以及相关命令；

变量定义 – 可定义命令表示一个字符串、一组编译命令或一组文件；

Makefile指示符 – 用于指示make工具需要执行的动作；

注释 – 以#表示注释，采用/#表示该字符。

# 2 make实际使用­­­­­­

对一个实际程序进行事例演示，main.cpp fun1.cpp fun2.cpp fun1.h fun2.h;

.\Makefile

#makefile for myapp

@echo “complie the project” #Tab后面表示shell命令（Linux）

.PHONY : clean #防止存在clean的文件

myapp : main.o fun1.o fun2.o

g++ main.o fun1.o fun2.o –o myapp

main.o : main.cpp fun1.h fun2.h

g++ -E main.cpp –o main.i

g++ -s main.i –o main.s

g++ -c main.s –o main.o # ?g++ -g -c main.s –o main.o 是否正确

fun1.o : fun1.cpp fun1.h

g++ -E fun1.cpp –o fun1.i

g++ -s fun1.i –o fun1.s

g++ -c fun1.s –o fun1.o

fun1.o : fun2.cpp fun2.h

g++ -E fun2.cpp –o fun2.i

g++ -s fun2.i –o fun2.s

g++ -c fun2.s –o fun2.o

clean :

$(RM) \*.i \*.s \*.o myapp #RM == rm –f == rm --force