4 规则书写

Makefile中的规则描述了何时以及如何重建特定的文件,这类特定的文件就被称为规则的目标(通常一条规则只有一个目标).规则中除了目标之外,还包含了目标的依赖文件,以及如果创建或更新该目标的配方.

除了 makefile 的默认目标”所在的规则,其他规则的书写顺序并不重要.默认目标就是当没有使用 make命令行指定具体目标时， make 默认的更新的那个目标.默认目标是Makefile中书写的第一条规则 中的目标.如果第一条规则中包含了多个目标,那么只有其中的第一个目标将被作为默认目标.这里存在两种例外情况: 一种是目标名以点号“.” 开始的并且其后不存在斜线“/”；另一种是模式规则中的目标.满足这两种情况之一的目标都不会被作为默认目标.

因此,通常情况下,我们在Makefile中书写的第一条规则就应该是编译整个程序或多个程序的最终描述(通常将这条规则的目标名写为"all").

4.1 规则语法

通常,一条规则的语法格式如下:

目标列表 : 依赖列表

配方

...

或者:

目标列表 : 依赖列表；配方

配方

...

"目标列表"通常是空格分开的多个文件名 .文件名除了可以使用通配符之外,格式"a(m)"表示档案文件a中的成员m.通常一条规则只包含一个目标,但偶尔情况下也会包含多个目标.

独立一行书写"配方"时,必须以一个tab字符开始(或者是由.RECIPEPREFIX变量指定的字符).第一条"配方"可以写在"依赖列表"后面并用分号";"隔开,并且多条配方也可以写在同一行,同样使用分号";"隔开即可.这些书写格式的效果都是一样的.

由于Makefile中使用"$"符号来引用变量,所以如果你想要在目标或依赖中单纯的使用"$"符号,需要写两个"$$".如果是在已经使能了二次展开功能的情况下,需要使用四个"$$$$"来达到同样的目的.

你可以使用反斜线"\"将一个较长的行分割成多个独立行,尽管这不是必须的,因为make程序对Makefile中一行的长度没有做限制.

一条规则告诉了make程序两件事:一件是目标何时过期；另一件是当需要时如何去更新目标.

目标是否过期的标准是由那些使用空格分割的"依赖列表"所决定的(通配符和档案文件成员在这里也同样适用).如果目标不存在或者它比"依赖列表"中的一些成员老(通过比较最后修改时间)时,则判定该目标过期.目标文件的内容是基于"依赖列表"的信息生成的,所以如果"依赖列表"中有成员发生变化,那么已经存在的目标文件必然不再有效.

更新目标的方法是由"配方"指定的,"配方"也就是一行或多行shell命令以及一些附加功能.

4.2 依赖类型

实际上有两种不同类型的依赖可以被GNU make程序理解: 一种是之前章节中描述的常规依赖；另一种是"order-only"依赖.常规依赖主要陈述了两件事:首先,由于目标的配方在执行之前必须确保该目标的所有常规依赖都已经更新,所以这些常规依赖的书写顺序决定了各自的更新顺序；其次,它确定了一个依赖关系,就是如果有任意一个常规依赖比目标新,那么就认为目标已经过期而需要被重建.

通常有一点是很明确的,那就是如果一个目标的依赖被更新了,则该目标也需要被更新.

但是,偶尔你会遇到这样一种情况,你想在规则的依赖列表中设置一个特殊的分类,这个特殊分类中的依赖被**更新**后不会强制规则的目标也进行更新.这种情况下,你就需要定义"order-only"依赖."order-only"依赖可以通过在依赖列表中放置管道符号"|"来进行设置: 管道符号“ |” 左边的是常规依赖， 管道符号右边的就是“order-only” 依赖.这样的规则书写格式如下：

目标列表 : 常规依赖部分 | "order-only"依赖部分

"常规依赖部分"可以为空.同样,也可以对一个目标进行多次追加依赖.需要注意的是,如果你在规则的常规依赖和"order-only"依赖中都设置了同一个文件,那么该文件将优先被视为常规依赖(因为常规依赖的行为是"order-only"依赖行为的一个超集).

考虑下这么一个例子,你的目标文件位于一个独立的目录下,而这个目录在执行make之前可能并不存在.在这种情况下,你希望只是在任何一个目标文件被放置进去之前创建该目录,但是因为目录的时间戳会因为里面任何一个文件被添加,删除,或者重命名而修改,我们显然不希望因为目录时间戳一改变就去重建目标文件.处理这个问题的一种方法就是使用"order-only"依赖: 将该目录设置为所有目标的"order-only"依赖.

OBJDIR := objdir

OBJS := $(addprefix $(OBJDIR)/,foo.o bar.o baz.o)

$(OBJDIR)/%.o : %.c

$(COMPILE.c) $(OUTPUT\_OPTION) $<

all: $(OBJS)

$(OBJS): | $(OBJDIR)

$(OBJDIR):

mkdir $(OBJDIR)

上面这个例子中,规则会在"objdir/foo.o"文件被创建之前先创建objdir这个目录,但没有"objdir/\*.o"文件会因为objdir目录时间戳变化而被重建.