2.5 Letting make Deduce the Recipes

It is not necessary to spell out the recipes for compiling the individual C source files, because make can figure them out: it has an implicit rule for updating a ‘.o’ file from a correspondingly named ‘.c’ file using a ‘cc -c’ command. For example, it will use the recipe ‘cc -c main.c -o main.o’ to compile main.c into main.o. We can therefore omit the recipes from the rules for the object files. See Using Implicit Rules.

没有必要详细说明编译单个 C 源文件的方法，因为 make 可以弄清楚它们：它有一个隐含的规则，使用 'cc -c' 命令从相应命名的 '.c' 文件更新 '.o' 文件。 例如，它将使用配方 “cc -c main.c -o main.o” 将 main.c 编译为 main.o。 因此，我们可以从目标文件的规则中省略配方。 请参阅使用隐式规则。

When a ‘.c’ file is used automatically in this way, it is also automatically added to the list of prerequisites. We can therefore omit the ‘.c’ files from the prerequisites, provided we omit the recipe.

当以这种方式自动使用 “.c” 文件时，它也会自动添加到先决条件列表中。 因此，只要我们省略了配方，我们就可以从先决条件中省略 “.c” 文件。

Here is the entire example, with both of these changes, and a variable objects as suggested above:

这是整个示例，包括这两个更改，以及上面建议的变量对象：

```
objects = main.o kbd.o command.o display.o \
          insert.o search.o files.o utils.o

edit : $(objects)
        cc -o edit $(objects)

main.o : defs.h
kbd.o : defs.h command.h
command.o : defs.h command.h
display.o : defs.h buffer.h
insert.o : defs.h buffer.h
search.o : defs.h buffer.h
files.o : defs.h buffer.h command.h
utils.o : defs.h

.PHONY : clean
clean :
        rm edit $(objects)
```

This is how we would write the makefile in actual practice. (The complications associated with ‘clean’ are described elsewhere. See Phony Targets, and Errors in Recipes.)

这就是我们在实际实践中编写 makefile 的方式。 （与 “clean” 相关的并发症在别处进行了描述。请参阅虚假目标和配方中的错误。）

Because implicit rules are so convenient, they are important. You will see them used frequently.

因为隐式规则如此方便，所以他们很重要。你会经常见到他们。
