2.4 Variables Make Makefiles Simpler

In our example, we had to list all the object files twice in the rule for edit (repeated here):

```
edit : main.o kbd.o command.o display.o \
              insert.o search.o files.o utils.o
        cc -o edit main.o kbd.o command.o display.o \
                   insert.o search.o files.o utils.o
```

Such duplication is error-prone; if a new object file is added to the system, we might add it to one list and forget the other. We can eliminate the risk and simplify the makefile by using a variable. Variables allow a text string to be defined once and substituted in multiple places later (see How to Use Variables).

这种重复很容易出错； 如果一个新的目标文件被添加到系统中，我们可能会将它添加到一个列表中而忘记另一个。 我们可以通过使用变量来消除风险并简化生成文件。 变量允许定义一次文本字符串，然后在多个位置替换（请参阅如何使用变量）。

It is standard practice for every makefile to have a variable named objects, OBJECTS, objs, OBJS, obj, or OBJ which is a list of all object file names. We would define such a variable objects with a line like this in the makefile:

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

每个 makefile 都有一个名为 objects、OBJECTS、objs、OBJS、obj 或 OBJ 的变量，这是所有对象文件名的列表，这是标准做法。 我们将在 makefile 中用这样的一行定义这样的变量对象：

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

Then, each place we want to put a list of the object file names, we can substitute the variable’s value by writing ‘$(objects)’ (see How to Use Variables).

然后，在每个我们想要放置目标文件名列表的地方，我们可以通过写 “$(objects)” 来替换变量的值（参见如何使用变量）。

Here is how the complete simple makefile looks when you use a variable for the object files:

```
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 : main.c defs.h
        cc -c main.c
kbd.o : kbd.c defs.h command.h
        cc -c kbd.c
command.o : command.c defs.h command.h
        cc -c command.c
display.o : display.c defs.h buffer.h
        cc -c display.c
insert.o : insert.c defs.h buffer.h
        cc -c insert.c
search.o : search.c defs.h buffer.h
        cc -c search.c
files.o : files.c defs.h buffer.h command.h
        cc -c files.c
utils.o : utils.c defs.h
        cc -c utils.c
clean :
        rm edit $(objects)
```