

## 目 录

<b>1</b>	<b>变量的基础</b>	<b>2</b>
<b>2</b>	<b>变量的高级用法</b>	<b>2</b>
2.1	变量值的替换 . . . . .	2
2.2	将变量的值再当成变量 . . . . .	2
<b>3</b>	<b>override 指示符</b>	<b>2</b>
<b>4</b>	<b>目标变量</b>	<b>3</b>

## 1 变量的基础

变量在声明时被赋予初值，而在使用时，需要在变量名前加上“\$”符号，并且用小括号“()”把变量包括起来。例子如下：

```
1  objects = program.o foo.o utils.o
2  program: $(objects)
3  cc -o program $(objects)
```

## 2 变量的高级用法

### 2.1 变量值的替换

通过格式“\$(var:a=b)”可以将变量“var”中所有以“a”字符串结尾的“a”替换为“b”字符串。例子如下：

```
1  foo := a.o b.o c.o
2  bar := $(foo:.o=.c)
3  # 此时$(bar)的值为“a.c b.c c.c”
```

### 2.2 将变量的值再当成变量

例子如下：

```
1  x = y
2  y = z
3  a := $( $(x) )
```

此时，\$(a) 的值为 z。

## 3 override 指示符

如果变量是通过 make 的命令行参数设置的，那么 makefile 中对这个变量的赋值将会被忽略。不过如果使用“override”指示符，就可以强行设置这类参数的值，语法如下：

```
1  override <variable> := <value>
```

## 4 目标变量

makefile 中可以为某个目标设置局部变量，它只在这条规则以及连带规则有效，这就是目标变量。

语法格式为：

```
1 <target ...>: <variable-assignment>
```

例子如下：

```
1  prog: CFLAGS = -g
2  prog: prog.o foo.o bar.o
3  $(CC)$(CFLAGS) prog.o foo.o bar.o
4
5  prog.o: prog.c
6  $(CC)$(CFLAGS) prog.c
7
8  foo.o: foo.c
9  $(CC)$(CFLAGS) foo.c
10
11 bar.o: bar.c
12 $(CC)$(CFLAGS) bar.c
```

在这个例子中，“CFLAGS” 的值是 “-g”。