目 录

1 函数的调用

函数的调用语法如下所示：

```
$(<function> <arguments>)
```

<function> 是函数名，<arguments> 是函数参数。函数名与函数参数用“空格”分隔，参数间用“,”分隔。函数的变量可以使用参数，例子如下：

```
$(subst a,b,$(x))
```

2 字符串处理函数

2.1 subst 函数

```
$(subst <from>,<to>,<text>)
# 功能：把字符串<text>中的<from>字符串替换为<to>字符串
# 返回：被替换后的字符串
```

2.2 patsubst

```
$(patsubst <pattern>,<replacement>,<text>)
# 功能：text中的单词用“空格”、“tab”、“回车”或“换行”分隔。如果单词符合模式pattern，就用replacement代替。pattern中可以有%，用于表示任意长度的字符串。如果replacement也有%，那么这里的%将是pattern中%代表的字符串。
# 返回：被替换后的字符串
```

2.3 strip

```
$(strip <string>)
# 功能：去掉string字符串中开头和结尾的空字符。
# 返回：被去掉空格的字符串
```

2.4 findstring

```
$(findstring <find>,<in>)
# 功能：在字符串<in>中查找<find>字符串
# 返回：如果找到，就返回find，否则返回空字符串。=
```

2.5 filter

```
$(filter <pattern>,<text>)
# 功能：保留text中符合pattern模式的单词。可以有多个模式，模式间用“空格”分隔。
# 返回：返回符合模式pattern的字符串
```

2.6 filter-out

```
$(filter-out <pattern>,<text>)
# 功能：删除text中符合pattern模式的单词。可以有多个模式，模式间用“空格”分隔。
# 返回：返回不符合模式pattern的字符串
```

2.7 sort

```
$(sort <list>)
# 功能：按字典序给list中的单词排序，单词间按“空格”分隔。sort函数还会去掉list
    中相同的单词。
# 返回：排序后的字符串。
```

2.8 word

```
$(word <n>,<text>)
# 功能：取出text中的第n个单词，从1开始。
# 返回：返回text中的第n个单词。如果n大于text的长度，则返回空字符串。
```

2.9 wordlist

```
$(wordlist <s>,<e>,<text>)
# 返回：text中从s到e的单词，从1开始。
```

2.10 words

```
$(words <text>)
# 返回：text中的单词数。
```

2.11 firstword

```
$(firstword <text>)
# 返回：text中的第一个单词。
```

3 文件名操作函数

3.1 dir

```
$(dir <names>)
# 返回：文件名的目录部分。如果没有反斜杠，就返回“./”
```

3.2 notdir

```
$(notdir <names>)
# 返回：文件名的非目录部分。
```

3.3 suffix

```
$(suffix <names>)
# 返回：各个文件名的后缀。如果文件没有后缀，就返回空字符串。
```

3.4 basename

```
$(basename <names>)
# 返回：各个文件名的前缀。如果文件没有前缀，就返回空字符串。
```

3.5 addsuffix

```
$(addsuffix <suffix>,<names>)
# 返回：将后缀suffix加到names中每个单词后面，然后返回结果。
```

3.6 addprefix

```
$(addprefix <prefix>,<names>)
# 返回：将前缀prefix加到names中每个单词前面，然后返回结果。
```

3.7 join

```
$(join <list1>,<list2>)
# 功能：将list2中每个单词对应地加到list1的单词后面。如果两者单词数量不一样，就
    将相应多出的单词保持原样。
```

4 foreach 函数

```
$(foreach <var>,<list>,<text>)
# 功能：将list中的单词放到var中指定的变量中，然后执行text中所包含的表达式，每一
    个text回返回一个字符串。
```

例子如下：

```
names := a b c d
files := $(foreach n,$(names),$(n).o)
```

需要注意的是，foreach 中的 var 是一个局部变量，其作用域中只在 foreach 函数中。

5 if 函数

```
$(if <condition>,<then-part>)
$(if <condition>,<then-part>,<else-part>)
# 如果else-part没有定义，那么当条件为假时，将返回空字符串。
```

6 call 函数

```
$(call <expression>,<param1>,<param2>,<param3>...)
# 功能：向表达式中传递参数
```

例子如下：

```
foo = $(call $(1) $(2),a,b)
# foo为a b
```

7 origin 函数

```
$(origin <variable>)
# 功能: 返回变量在哪里定义的。
# 返回值:
# undefined, 变量未定义
# default, 变量是一个默认的定义
# environment, 变量是环境变量
# file, 变量定义在makefile中
# command line, 变量由命令行定义
# override, 变量被override指示符重新定义过
# automatic, 变量是一个自动化变量
```

8 shell 函数

```
$(shell command)
# 功能: 返回命令行的输出
```

例子如下:

```
files := $(shell echo *.c)
```

9 控制 make 的函数

9.1 error

```
$(error <text>)
# 功能: 产生一个错误, text是错误信息, 输出text后, 终止make的运行。
```

9.2 warning

```
$(warning <text>)
# 功能: text是警告信息, 输出警告信息后, 让make继续运行。
```