

写在前面的话

本节视频来自2018年为2440的裸机加强版视频，非常适合本教程。

要注意几点：

- 以前源码目录是 003_Makefile，现在目录改为 04_2018_Makefile
 - GIT仓库里：01_all_series_quickstart\04_嵌入式Linux应用开发基础知识\source\04_2018_Makefile
- 本节视频配套的文档，就是本文档，位于：
 - GIT仓库里：01_all_series_quickstart\04_嵌入式Linux应用开发基础知识\doc_pic\04.2018_Makefile

Makefile函数

makefile里面可以包含很多函数，这些函数都是make本身实现的，下面我们来几个常用的函数。引用一个函数用“\$”。

函数foreach

函数foreach语法如下：

```
$(foreach var,list,text)
```

前两个参数，‘var’和‘list’，将首先扩展，注意最后一个参数‘text’此时不扩展；接着，对每一个‘list’扩展产生的字，将用来为‘var’扩展后命名的变量赋值；然后‘text’引用该变量扩展；因此它每次扩展都不相同。结果是由空格隔开的‘text’。在‘list’中多次扩展的字组成的新的‘list’。‘text’多次扩展的字串联起来，字与字之间由空格隔开，如此就产生了函数foreach的返回值。

实例：

```
A = a b c
B = $(foreach f, &(A), $(f).o)

all:
    @echo B = $(B)
```

结果：

```
B = a.o b.o c.o
```

函数filter/filter-out

函数filter/filter-out语法如下：

```
$(filter pattern...,text)      # 在text中取出符合patten格式的值
$(filter-out pattern...,text)  # 在text中取出不符合patten格式的值
```

实例：

```
C = a b c d/

D = $(filter %/, $(C))
E = $(filter-out %/, $(C))

all:
    @echo D = $(D)
    @echo E = $(E)
```

结果:

```
D = d/
E = a b c
```

Wildcard

函数Wildcard语法如下:

```
$(wildcard pattern) # pattern定义了文件名的格式，wildcard取出其中存在的文件。
```

这个函数 wildcard 会以 pattern 这个格式，去寻找存在的文件，返回存在文件的名字。

实例:

在该目录下创建三个文件: a.c b.c c.c

```
files = $(wildcard *.c)

all:
    @echo files = $(files)
```

结果:

```
files = a.c b.c c.c
```

我们也可以用wildcard函数来判断，真实存在的文件

实例:

```
files2 = a.c b.c c.c d.c e.c abc
files3 = $(wildcard $(files2))

all:
    @echo files3 = $(files3)
```

结果:

```
files3 = a.c b.c c.c
```

patsubst函数

函数 patsubst 语法如下:

```
$(patsubst pattern,replacement,\$(var))
```

patsubst 函数是从 var 变量里面取出每一个值，如果这个符合 pattern 格式，把它替换成 replacement 格式，

实例：

```
files2 = a.c b.c c.c d.c e.c abc  
  
dep_files = $(patsubst %.c,%.d,$(files2))  
  
all:  
    @echo dep_files = $(dep_files)
```

结果：

```
dep_files = a.d b.d c.d d.d e.d abc
```