

## shui1025701856的专栏

- [目录视图](#)
- [摘要视图](#)
- [订阅](#)

赠书 | 异步2周年,技术图书免费选  
畅

每周荐书：分布式、深度学习算法、iOS（评论送书）

项目管理+代码托管+文档协作，开发更流

## makefile 自动编译同一个目录下的所有文件

标签：[makefilewildcard](#)[编译器](#)[includemathshell](#)

2012-04-14 11:52 10848人阅读 [评论\(1\)](#) [收藏](#) [举报](#)

分类：

嵌入式linux高级编程学习 (11)

[作者同类文章](#)X



版权声明：本文为博主原创文章，未经博主允许不得转载。

```
[csharp]

01.
02. SOURCE = $(wildcard *.c)
03. OBJS = $(patsubst %.c,%.o,$(SOURCE))
04.
05. CROSS_COMPILE = arm-linux-
06. CXX = gcc
07.
08. CFLAGS += --static
09. CLFAGS += -Wall
10. LDFLAGS += -lm
11.
12. all:bts
13. bts:$(OBJS)
14.     $(CROSS_COMPILE)$(CXX) -Wall $(CFLAGS) -o $@ $^
15.
16. .PHONY:clean
17. clean:
18.     rm -f *.o *.d bts
19.
20. include $(SOURCE:.c=.d)
21.
22. %.o:%.c
23.     $(CROSS_COMPILE)$(CXX) -Wall $(CFLAGS) -c $< -o $@
24.
25. %.d: %.c
26.     @set -e; rm -f $@; \
27.     $(CXX) -MM $(CFLAGS) $< > $@.

; \
28.     sed 's,$$* \\.o[ :]*,\\1.o $@ : ,g' < $@.

> $@; \
29.     rm -f $@.
```

1.SOURCE=\$(wildcard \*.c) //返回值为当前目录下所有.c源文件列表

`$(wildcard PATTERN)`

函数名称：获取匹配模式文件名函数-wildcard 函数功能：列出当前目录下所有符合模式“PATTERN”格式的文件名。

返回值：空格分割的、存在当前目录下的所有符合模式“PATTERN”的文件名。函数说明：“PATTERN”使用shell可识别的通配符，包括“?”（单字符）、“\*”（多字符）等。示例：

`$(wildcard *.c)` 返回值为当前目录下所有.c源文件列表。

`$(patsubst PATTERN,REPLACEMENT,TEXT)` 函数名称：模式替换函数—patsubst。

2.OBJs = `$(patsubst %.c,%.o,$(SOURCE))` //把字符串\$(SOURCE)中以.c结尾的单词替换成以.o结尾的字符。函数的返回结果是.o的文件

`$(patsubst PATTERN,REPLACEMENT,TEXT)`

函数名称：模式替换函数—patsubst。

函数功能：搜索“TEXT”中以空格分开的单词，将符合模式“PATTERN”替换为“REPLACEMENT”。参数“PATTERN”中可以使用模式通配符“%”来代表一个单词中的若干字符。如果参数“REPLACEMENT”中也包含一个“%”，那么“REPLACEMENT”中的“%”将是“PATTERN”中的那个“%”所代表的字符串。

在“PATTERN”和“REPLACEMENT”中，只有第一个“%”被作为模式字符来处理，之后出现的不再作模式字符（作为一个字符）。在参数中如果需要将第一个出现的“%”作为字符本身而不作为模式字符时，可使用反斜杠“\”进行转义处理（转义处理的机制和使用静态模式的转义一致）

返回值：替换后的新字符串。函数说明：参数“TEXT”单词之间的多个空格在处理时被合并为一个空格，并忽略前导和结尾空格。示例：

`$(patsubst %.c,%.o,x.c bar.c)` 把字符串“x.c bar.c”中以.c结尾的单词替换成以.o结尾的字符。函数的返回结果是“x.o bar.o”

3.CROSS\_COMPILE = //交叉编译变量

4.CXX = gcc //gcc为C语言编译器、g++为c++编译器

5.CFLAGS += -D\_\_DEBUG\_\_ //+= 向后添加有空格。增加调试信息D\_\_DEBUG\_\_

6.CLFAGS += -Wall //增加警告信息

7.LDFLAGS += -lm //增加链接到math库

8.all:bts //指定执行一个由伪目标定义的若干条命令或者一个空目标文件all 作为Makefile的顶层目标，一般此目标作为默认的终极目标。

9.bts:\$(OBJs) //bits是终极目标它依赖于所有的.o文件

10.\$(CROSS\_COMPILE)\$(CXX) -Wall \$(LDLAGS) -o \$@ \$^

//根据实际的编译器选择变量-o \$@：生成目标文件的完整名称

\$^:依赖所有的依赖\$(OBJs)

11..PHONY:clean

clean:

rm -f \*.o \*.d bts //定义伪目标删除所有的.o .d 和生成的执行目标。

所有的.d文件依赖于同名的.c文件。.d文件里包含了.c所依赖的头文件信息

将一个目标声明为伪目标的方法是将其作为特殊目标.PHONY”的依赖。PHONY: clean 这样目标“clean”就被声明为一个伪目标，无论在当前目录下是否存在“clean”这个文件。我们输入“make clean”之后。“rm”命令都会被执行。而且，当一个目标被声明为伪目标后，make在执行此规则时不会去试图去查找隐含规则来创建它。这样也提高了make的执行效率，同时也不用担心由于目标和文件名重名而使我们的期望失败。在书写伪目标规则时，首先需要声明目标是一个伪目标，之后才是伪目标的规则定义。目标“clean”的完整书写格式应该如下：.PHONY: clean

clean:

rm \*.o temp

12.include \$(SOURCE:.c=.d) //删除临时文件:在 Makefile 中书写一个伪目标“depend”的规则来定义自动产生依赖关系文件的命令。输入“make depend”将生成一个称为“depend”的文件，其中包含了所有源文件的依赖规则描述。Makefile中使用“include”指示符包含这个文件。Makefile中对当前目录下.d文件处理可以参考如下：sources = foo.c bar.c sinclude \$(sources:.c=.d) 例子中，变量“sources”定义了当前目录下的需要编译的源文件。变量引用置换“\$(sources:.c=.d)”的功能是根据变量“source”指定的.c文件自动产生对应的.d文件，并在当前Makefile文件中包含这些.d文件。

13.%o:%c

```
$(CROSS_COMPILE)$(CXX) -Wall $(CFLAGS) -c $< -o $@
```

//模式规则的格式为:            %o : %c ; COMMAND...

此规则描述了一个.o文件如何由对应的.c文件创建。

首先看编译.c文件到.o文件的隐含模式规则:

```
%o : %c
```

\$(CC) -c \$(CFLAGS) \$(CPPFLAGS) \$< -o \$@ 规则的命令行中使用了自动化变量“\$<”和“\$@”，其中自动化变量“\$<”代表规则的依赖，“\$@”代表规则的目标。此规则在执行时，命令行中的自动化变量将根据实际的目标和依赖文件取对应值

14.%d: %c

```
@set -e; rm -f $@; \
```

```
$(CXX) -MM $(CFLAGS) $<> $@.
```

```
; \
```

```
sed 's,$ * \.o[:]*,l.o $@: ,g' < $@.
```

```
> $@; \
```

```
rm -f $@.
```

//自动生成每一个.c文件对应的.d文件.此规则的含义是：所有的.d文件依赖于同名的.c文件。

顶

0

踩

0

-

- 上一篇[V4L2驱动程序架构](#)
- 下一篇[android camera HAL](#)

## 相关文章推荐

- [makefile 自动编译同一个目录下的所有文件](#)
- [POCO库中文编程参考指南（10）如何使用TCPServer框架？](#)
- [一个简单的makefile的实现，编译当前目录下所有的.c文件](#)
- [两个简单的makefile的实现，编译当前目录下所有的.c文件](#)
- [多文件目录下makefile文件递归执行编译所有c文件](#)
- [makefile多目录的.c 格式.cpp混合编译](#)
- [多文件目录下makefile文件递归执行编译所有c文件](#)
- [多文件目录下makefile文件递归执行编译所有c文件（小例子1）](#)
- [Makefile编译目录下多个文件](#)
- [Makefile自动编译目录下的所有.C文件](#)

