**参考文档：**<https://www.jianshu.com/p/7e49071f9e11>

### Makefile 的基本工作原理

makefile默认执行第一个目标

make XX的时候，Makefile会自动执行xx这个目标下面的命令语句。

当我们make xx的时候，是否执行命令是取决于依赖的。依赖如果成立就会执行命令，否则不执行。

### Makefile 的工作方式

读入主Makefile（主Makefile中可以引用其他Makefile）

读入被include的其他Makefile

初始化文件中的变量

推导隐晦规则, 并分析所有规则

为所有的目标文件创建依赖关系链

根据依赖关系, 决定哪些目标要重新生成

执行生成命令

### Makefile的基本语法知识

#### Makefile 中定义和使用变量

**定义变量（=or:=）**

其中 = 和 := 的区别在于, := 只能使用前面定义好的变量, = 可以使用后面定义的变量

#此行为注释行

#测试变量定义 =

OBJS2 = $(OBJS1) c.c

OBJS1 := $(OBJS0) b.c

OBJS0 = a.c

all:

    @echo $(OBJS2)

**输入 make test1**

**输出结果 b.c c.c**

**变量替换**

#变量替换

SRCS := a.c b.c c.c

OBJS := $(SRCS:%.c=%.o)

test2:

    @echo "SRCS: " $(SRCS)

    @echo "OBJS: " $(OBJS)

**输入 make test2**

**输出结果 SRCS: a.c b.c c.c**

**OBJS: a.o b.o c.o**

**变量追加**

SRCS1 := a.c b.c c.c

SRCS1 += d.c

test3:

    @echo "SRCS: " $(SRCS1)

**输入 make test3**

**输出结果 SRCS: a.c b.c c.c d.c**

**变量覆盖**

#变量覆盖

SRCS2 := d.c

override SRCS2 := a.c b.c c.c

test3:

    @echo "SRCS: " $(SRCS2)

**输入make test4**

**输出结果 SRCS: a.c b.c c.c**

**目标变量**

作用：使变量的作用域仅限于这个目标(target)

语法：<target ...> : <variable-assignment>

test5: TARGET1-SRCS := d.c

test5:

    @echo "SRCS: " $(TARGET1-SRCS)

test6:

    @echo "SRCS: " $(TARGET1-SRCS)

**输入：make test5**

**输出结果：SRCS: d.c**

**输入：make test6**

**输出结果：SRCS:**

#### Makefile 命令前缀

Makefile 中书写shell命令时可以加2种前缀 @ 和 -, 或者不用前缀

* 不用前缀 ：输出执行的命令以及命令执行的结果, 出错的话停止执行
* 前缀 @ ：只输出命令执行的结果, 出错的话停止执行
* 前缀 - ：命令执行有错的话, 忽略错误, 继续执行
* test7:
* echo "没有前缀"
* cat this\_file\_not\_exist
* echo "错误之后的命令"
* test8:
* @echo "前缀@"
* @cat this\_file\_not\_exist
* @echo "错误之后的命令"
* test9:
* -echo "前缀-"
* -cat this\_file\_not\_exist
* -echo "错误之后的命令"

**输入make test7:**

**输出结果echo "没有前缀"**

**没有前缀**

**cat this\_file\_not\_exist**

**cat: this\_file\_not\_exist: 没有那个文件或目录**

**Makefile:35: recipe for target 'test7' failed**

**make: \*\*\* [test7] Error 1**

**输入make test8:**

**输出结果：前缀@**

**cat: this\_file\_not\_exist: 没有那个文件或目录**

**Makefile:35: recipe for target 'test8' failed**

**make: \*\*\* [test8] Error 1**

**输入make test9:**

**输出结果：echo "前缀-"**

**前缀-**

**cat this\_file\_not\_exist**

**cat: this\_file\_not\_exist: 没有那个文件或目录**

**Makefile:43: recipe for target 'test9' failed**

**make: [test9] Error 1 (已忽略）**

**echo "错误之后的命令"**

**错误之后的命令**

#### Makefile 的伪目标（.PHONY）

(1)伪目标意思是这个目标本身不代表一个文件，执行这个目标不是为了得到某个文件或东西，而是单纯为了执行这个目标下面的命令。

(2)伪目标一般都没有依赖，因为执行伪目标就是为了执行目标下面的命令。既然一定要执行命令了那就不必加依赖，因为不加依赖意思就是无条件执行。

(3)伪目标可以直接写，不影响使用；但是有时候为了明确声明这个目标是伪目标会在伪目标的前面用.PHONY来明确声明它是伪目标。

典型的伪目标是 Makefile 中用来清理编译过程中中间文件的 clean 伪目标, 一般格式如下:

.PHONY: clean   <-- 这句没有也行, 但是最好加上

clean:

    -rm -f \*.o

#### 引用其他的 Makefile

语法: include <filename> (filename 可以包含通配符和路径)

./src/Makefile:

src-test1:

    @echo "other makefile begin"

    @echo "other makefile end"

./makefile

include ./src/Makefile

test10:

    @echo "主makefile begin"

    @make src-test1

    @echo "主makefile end"

输入：make test10

输出结果：

主makefile begin

make[1]: 进入目录“/home/liuxiang/WorkeSpace/Demo\_Makefile/Demo\_Makefile”

other makefile begin

other makefile end

make[1]: 离开目录“/home/liuxiang/WorkeSpace/Demo\_Makefile/Demo\_Makefile”

主makefile end