

## 目 录

<b>1</b>	<b>makefile 的组成</b>	<b>2</b>
<b>2</b>	<b>引用其他的 makefile</b>	<b>2</b>
<b>3</b>	<b>make 的工作方式</b>	<b>2</b>

## 1 makefile 的组成

makefile 由以下 5 部分组成：

- 显式规则。显式规则用于说明要生成的文件、文件的依赖文件和生成的命令。
- 隐含规则。当我们比较简略地书写 makefile 时，可以自动推导其中的功能。
- 变量的定义。类似于 C 语言的宏，在 makefile 被执行时，其中的变量会被扩展到相应的引用位置上。
- 文件指示。文件指示包括引用其他 makefile 文件、指定 makefile 中的有效部分和定义多行的命令。
- 注释。使用 “#” 进行代码的注释。

## 2 引用其他的 makefile

在 makefile 中使用 include 关键字可以引用其他的 makefile 文件，语法如下：

```
1 include <filename>
```

需要注意的是，include 关键字前面可以有空格，但是不可以有 tab 键。

include 查找引用文件的方式如下：

1. 当 filename 有指定绝对路径或是相对路径时，将直接寻找路径处的 file。
2. 当 filename 没有指定绝对路径或是相对路径时，将首先在当前目录下寻找 file。如果当前目录下没找到，make 还将到由 “-include-dir” 参数所指定的目录下去寻找。如果仍然没找到，或者未指定 “-include-dir” 参数，将在 “<prefix>/include” 目录（一般为：/usr/local/bin 或/usr/include）下去查找。
3. 如果以上两种方式都没找到引用的 makefile，make 就会报错。如果想让 make 不理睬那些无法读取的文件继续执行，可以在 include 前加一个 “-”，命令如下：

```
1 -include <filename>
```

## 3 make 的工作方式

make 工作分为 7 个步骤，如下所示：

1. 读入所有的 makefile。

2. 读入被 include 的其他 makefile。
3. 初始化文件中的变量。
4. 推导隐晦规则，并分析所有规则。
5. 为所有的目标文件创建依赖关系链。
6. 根据依赖关系，决定哪些目标要重新生成。
7. 执行生成命令。