

## MakeFile总结

```
target ... : prerequisites ...  
    command  
    ...  
    ...
```

- target: 要生成的目标文件。
- prerequisites: 要生成那个target所需要的文件或是目标。
- prerequisites中如果有一个以上的文件比target文件要新的话, 或者target不存在的话, command所定义的命令就会被执行。
- command: make需要执行的命令。(任意的shell命令)

target这一个或多个的目标文件依赖于prerequisites中的文件, 其生成规则定义在 command中。

- clean不是一个文件, 它只不过是一个动作名字, 类似于C语音中的label, 其冒号后什么也没有, 即没有任何依赖prerequisites。
  - make就不会自动去找它的依赖性, 也就不会自动执行其后所定义的命令。
  - 要执行其后的命令(不仅用于clean, 其他lable同样适用), 就要在make命令后明显得指出这个lable的名字。

### • make的工作流程

1. make会在当前目录下找名字叫“Makefile”或“makefile”的文件。
2. 如果找到, 它会找文件中的第一个目标文件(target)。
  1. 如果目标文件(target)不存在, 或是其所依赖的后面的.o文件的文件修改时间要比target这个文件新, 那么, 他就会执行后面所定义的命令来生成target这个文件。
3. 如果目标文件(target)所依赖的.o文件也不存在, 那么make会在当前文件中找目标为.o文件的依赖性, 如果找到则再根据那一个规则生成.o文件。
4. make会一层又一层地去找文件的依赖关系, 直到最终编译出第一个目标文件。
  1. make只管文件的依赖性, 依赖的文件找不到, 那么make就会直接退出, 并报错。而对于编译错误等, make根本不理, 继续执行。

### • make 特性

- make自动推导
  - 可以自动推导文件以及文件依赖关系后面的命令。
    - 只要make看到一个['name'.o]文件, 它就会自动的把相同文件名的['name'.c]文件加在依赖关系中
- make命令会在当前目录下按顺序找寻文件名为“GNUmakefile”、“makefile”、“Makefile”的文件。
- 引用其它的Makefile

```
include <filename>;  
include filename
```

- make命令开始时，会找寻include所指出的其它Makefile，并把其内容安置在当前的位置。
- 如果文件都没有指定绝对路径或是相对路径的话，make会在当前目录下首先寻找，如果当前目录下没有找到，那么，make还会在下面的几个目录下找
  - 如果make执行时，有“-I”或“--include-dir”参数，那么make就会在这个参数所指定的目录下去寻找。
  - 如果目录<prefix>:/include（一般是：/usr/local/bin或/usr/include）存在的话，make也会去找。
- .PONY 将之后的label设置为伪命令。