

makefile中伪目标的两个典型应用

makefile中目标的三个规则

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

- target

可以是一个object file（目标文件），也可以是一个执行文件，还可以是一个标签（label）。对于标签这种特性，在后续的“伪目标”章节中会有叙述。

- prerequisites

生成该target所依赖的文件和/或target

- command

该target要执行的命令（任意的shell命令）

伪目标

```
.PHONY : pseudo_target
```

使用 `.PHONY` 命令来声明一个目标为伪目标，伪目标可以没有 prerequisites 部分或 command 部分，但是一旦声明了这两个部分，他们就会按照目标的规则来执行。

用伪目标实现同时生成多个可执行文件

```
all : prog1 prog2 prog3
.PHONY : all

prog1 : prog1.o utils.o
      cc -o prog1 prog1.o utils.o

prog2 : prog2.o
      cc -o prog2 prog2.o

prog3 : prog3.o sort.o utils.o
      cc -o prog3 prog3.o sort.o utils.o
```

这里用到了伪目标的prerequisites原则，声明all为伪目标，这里没有定义all的command部分，但是要求all必须依赖于prog1 prog2 prog3三个可执行程序，因此这三个都会被编译连接生成，但是all是一个伪目标，伪目标并不会被真正生成。这里相当于利用prerequisits原则同时生成多个可执行文件。

用伪目标进行清理

```
.PHONY : clean
clean :
      rm *.o target
```

这里用到了伪目标的command原则，声明clean为伪目标，这里没有定义clean的prerequisites部分，因此在执行 `make clean` 时，就相当于运行clean的command部分，没有什么依赖。因此可以用这个原则去进行清理文件。

同时利用两个原则进行中间文件清理

```
all : target1 target2 cleanobj
.PHONY : all

.PHONY : cleanobj
cleanobj :
    rm *.o
```

声明all伪变量的prerequisites部分为target1 target2 cleanobj，这时要求生成这三个目标，其中会生成target1 target2的可执行文件，又一定会运行cleanobj的command部分，进行中间文件的清理。在运行make命令时就会执行清理动作，不需要单独运行make clean。