

# Makefile

**基本原理:** Makefile是一个文本形式的数据库文件，其中包含一些规则来告诉make处理哪些文件以及如何处理这些文件。规则主要描述哪些文件是从哪些文件中产生，以及用什么命令来执行这个过程。依靠规则的描述信息，make对磁盘上的文件进行检查，如果目标文件的生成或被改动的时间（时间戳）至少比它的一个依赖文件还旧，make就执行相应命令，以更新目标文件。

## Makefile基本语法:

target dependency dependency

(tab) <command>

**注:** 目标可以是文件，也可以是执行的动作（如clean）

## 一个最简单原始的Makefile:

```
prog:prog1.o prog2.o
gcc prog1.o prog2.o -o prog
prog1.o:prog1.c lib.h
gcc -c -l. -o prog1.o prog2.c
prog2.o:prog2.c
gcc -c prog2.c
```

**问题:** 上面的Makefile过于麻烦，如果打算换一个编译器，就要把所有gcc替换为arm-linux-gcc;

**解决:** 定义与使用

**定义:** VARNAME=string

**使用:** \${VARNAME}

## 简化后的Makefile:

```
OBJS=prog.o code.o
CC=gcc
test:${OBJS}
```

```
    ${CC} -o test ${OBSJ}
prog.o:prog.c prog.h code.h
    ${CC} -c prog.c -o prog.o
code.o:code.c code.h
    ${CC} -c code.c -c code.o
clean:
    rm -f *.o
```

Makefile中常见的三个符号，\#@代表目标文件，\%^代表所依赖的文件，\$<代表第一个依赖的文件

**例如:**

```
//arm-linux-ld      是链接命令
//arm-linux-objcopy 复制文件进行格式转换
//arm-linux-objdump 反汇编
led.bin:start.o
    arm-linux-ld -Ttext 0x0 -o led.elf ^
    arm-linux-objcopy -O binary led.elf led.bin
    arm-linux-objdump -D led.elf > led_elf.dis
    gcc mkv210_image.c -o mkmini210
    ./mkmini210 led.bin 210.bin
%.o:%.s
    arm-linux-gcc -o $@ $< -c
%.o:%.c
    arm-linux-gcc -o $@ $< -c
```

**结尾:**

初学Makefile将其分段整理成笔记供自己参考也供与大家学习，如有错误请大佬们直言指出，如果感觉有用那就点个赞留个言，谢谢观众老爷们的赏脸。

若想获得上述内容的PDF版本移步到GitHub下载。

**地址:** <https://github.com/QianquanChina/Study-Notes>

